A Survey of Knowledge Acquisition from Natural Language[1]

Stephen Potter,
Artificial Intelligence Applications Institute,
Division of Informatics,
University of Edinburgh,
80 South Bridge,
Edinburgh, EH1 1HN

email: stephenp@aiai.ed.ac.uk

1 Introduction

Knowledge Acquisition (KA) is the process of acquiring (either directly from a human or from some other source) that information, and its formalised structure, that will allow some particular task to be performed by a computer system. In AI, structured information of this sort is commonly termed 'knowledge', and it is in this sense that the term is used throughout this document.

In this survey, consideration is given to KA from Natural Language (NL), in other words, the acquisition of knowledge from some (spoken or written) source expressed in an everyday human language - this would encompass approaches to KA such as interrogating human experts and reading textbooks. This might be contrasted to an approach such as asking experts to express their knowledge 'directly' in some formal (and non-natural) manner – say, by writing down a set of rules - or to an approach such as machine-learning knowledge from a set of examples of the expert's behaviour, in which the knowledge is expressed indirectly through these examples.

The survey covers three principal subject areas. The following section attempts to present a very brief overview of the current 'conventional' approaches of KA from NL. Primarily, these have been developed or appropriated with the express purpose of acquiring knowledge for intelligent systems from NL communication. They are the standard tools used in the construction of such systems, and generally involve human-human interaction with computers used in a support role, if at all. The section concludes with a short discussion of the general problems encountered when performing KA in this fashion.

Section 3 contains a description of some of the techniques that have been developed with the intention of introducing some degree of automation into the process of KA from NL. As will be seen, these approaches are diverse in nature, and, on the whole, lack the maturity necessary for practical application.

These techniques often use ideas (and, occasionally, software) developed in the field of Language Engineering (LE). In general, research in this field is concerned with attempts to automate some aspect of human communication, and as

[1] This document represents a *Technology Maturity Assessment* (TMA) of the field of Knowledge Acquisition from Natural Language. This work has been produced as part of the Advanced Knowledge Technologies project, Work Package 1, AKT challenges, subpackage 1.1, knowledge acquisition, task 1.1.2, KA and natural language.

such its focus lies more upon the NL element. However, while not designed expressly for this purpose, LE techniques often have the potential to be applied to KA tasks or to assist in the KA process. Consequently, section 4 contains a survey of this field, along with an attempt to convey the current state of the art of the various technologies to be found there.

2 Conventional Knowledge Acquisition from Natural Language

Typically, the process of KA for some purpose will be performed by a human *knowledge engineer*, who is familiar with the techniques for representing knowledge on computer, and is able to translate the information that he/she receives about the task into an appropriate form. At the outset of the process, he/she will not necessarily be familiar with the task in hand, or the working domain, but will often need to acquire an understanding of at least the principal concepts of both domain and task in order to perform the KA satisfactorily. The knowledge may be derived from a number of sources - textbooks, archived material, on-line documentation, etc. - but at some point in most conventional KA processes there will be a need to interact in some manner with one or more human task experts in order to try to access some elements of the required knowledge.

As mentioned above, for the purposes of this review, interest lies in techniques for KA from natural language; in other words, techniques for extracting this structured information from written or spoken examples of 'normal' communication. Consequently, this excludes consideration of 'contrived' techniques for KA; using such methods, an attempt is made to access the expert's knowledge using artificial prompts (for instance, a set of concept cards, which the expert might be asked to arrange into a form that mirrors the structure of the domain). When using such techniques, there should be little or no need to analyse natural language, since they are used with the express intention of making the knowledge explicit through the words or actions of the expert.

Of those techniques that do involve KA from NL, many are informal (such as the chats between the knowledge engineer and an expert over coffee) or otherwise difficult to quantify (such as the knowledge engineer's study of domain textbooks). Particularly during the preliminary stages of KA, techniques of this sort will often be useful (and perhaps even necessary) for allowing the knowledge engineer to acquire background knowledge of the task and also for creating a social climate to facilitate the rest of the KA process. However, their nature means that it is practically impossible to assess the range and success of such techniques.

There is a vast literature devoted to more formal (manual) techniques for capturing the knowledge contained within NL; a discussion of these techniques is considered to be beyond the scope of the current report, in which the emphasis lies on KA *technology* maturity (and, hence, the use of techniques involving some degree of automation). However, in order to introduce the current issues and problems with KA from NL, two manual techniques of a general nature – namely, *interviews* and *Protocol Analysis* - will now be briefly discussed, and their limitations outlined.

2.1 Interviews

An interview between a human expert and a knowledge engineer is probably the most common form of knowledge acquisition. Interviews are usually taped and later transcribed and analysed. They may either be unstructured or else have a more formal nature. Unstructured interviews are often used during the early stages of a KA process to gain an overview of the essentials of the domain without probing too deeply into specific topics. The disadvantage of the unstructured approach lies in its very nature; the lack of constraint and formality means that the interview may stray

from the subject, the expert may offer incomplete or contradictory information, and the analysis of the resulting transcription for useful knowledge may be a thankless task.

Structured interviews, on the other hand, are more rigorously planned, and are more tightly controlled by the interviewer, and often have the acquisition of some particular scrap of knowledge as their goal (and, as such, are often used with the intention of plugging gaps in the knowledge base). Since they are more structured, the resulting transcriptions should also be of a more structured nature which should facilitate their analysis. However, they do require both quite detailed preparation and a skilled interviewer.

(A different approach to the acquisition of knowledge - of a particular type - from interviews can be found in the technique of discourse analysis (Belkin, Brooks and Daniels, 1988). This technique has been developed expressly for the purpose of acquiring knowledge for building intelligent interfaces, and is based on the analysis of typical dialogues between experts and their customers; the focus of the KA exercise surrounds this interview process itself; the dialogues are analysed to try to determine the form and content of both the questions asked and the corresponding replies. This analysis is then used to develop the interaction mechanisms of the interface.)

Whether structured or not, however, there will usually be a need for a number of interviews, and the extraction of knowledge in this way can be slow and laborious. The interview approach relies on the existence of a willing, able and articulate expert. If, for whatever reason, such an expert is not available, then the process will be of limited usefulness. Another factor is the ability and knowledge of the knowledge engineer: without a sufficient familiarity with the terminology and concepts of the domain, he/she may find it difficult to control the process. In addition to the problems suggested above, a number of researchers (e.g., see (Berry, 1987), (Hart, 1988), (Gillies, 1996)) have highlighted the problem of *compiled* or *tacit* knowledge: the interview technique relies upon the expert's ability to verbalise his/her knowledge. However, it is often the case that there will be some knowledge that even the most articulate of experts will not (or will not be able to) express verbally. This might be because it is implicit within the domain, or because it is non-verbal in nature (pattern recognition knowledge, for instance) or else because it has been 'compiled' through experience and repeated use, and cannot be retrieved.[2]

2.2 Protocol Analysis

Protocol Analysis (PA) (Ericsson and Simon, 1984) is the term given to a number of techniques which present the expert with a task to perform, and while doing it, require him/her to try to 'think aloud' and articulate the stages in reasoning and the focus of attention at every point in the process. (The knowledge engineer may also be recording the expert's actions.) The expert's self-reporting may occur 'on-line' while the task is being performed, or retrospectively, providing a commentary to a recording of the problem-solving session. This self-report or commentary provides the 'protocol' to be analysed; once it has been transcribed, the analysis can begin.

The goal of the analysis is to identify 'chunks' of knowledge (which may represent anything from domain concepts to high-level general problem-solving approaches); these may be indicated by the presence of syntactical cues (for

[2] Naturally, since it is a constraint on language, this problem may be encountered during any attempt to acquire knowledge from NL, regardless of the particular acquisition technique adopted. This should be borne in mind throughout the course of the discussion in this document.

example, concepts are likely to be nouns, and (sub-)tasks/processes verbs) or other linguistic pointers. This approach suffers from many of the problems associated with the interview technique as described above; additional difficulties particular to PA include (Schreiber et al., 1999):

- choosing the right example task — since each PA episode is time-consuming and difficult, it is likely that there will be few opportunities to apply this technique in the course of a KA process. As a consequence, the importance of choosing appropriate tasks for the expert to perform is heightened. The tasks should be neither trivial nor too difficult, and (usually) representative of a typical task (although, in certain circumstances, choosing atypical or unusual tasks can give a clearer insight into the expert's working).
- it might be the case that the act of self-reporting actually interferes with the expert's reasoning, and so the protocol gained will not be representative at all.
- the artificial nature of the task and its context (obviously, the experts are aware that they are being observed) may hinder or alter the expert's approach to it.
- the protocol may include irrelevant – but not immediately obvious as such - utterances which arise when the expert comments on things other than the task in hand; this can complicate the analysis procedure.

2.3 Summary

Much of the knowledge for intelligent systems is acquired via the medium of NL. This is perhaps unsurprising when one considers that intelligent systems are usually intended to replicate some human problem-solving ability, and NL is the obvious means by which humans can express how the task is performed - the most natural approach to KA is to ask human experts for their knowledge.

Although successful intelligent systems have been constructed using knowledge acquired in this fashion, the conventional approaches to KA from NL are not without their weaknesses; these may be classified as their *theoretical* and their *practical* limitations.

2.3.1 Theoretical Limitations

Attempts to acquire knowledge from NL assume that the required knowledge can be expressed in this form. However, it seems as though certain aspects of human expertise cannot be expressed in this way. For example, complex pattern-matching knowledge, which might be crucial to the task in hand, cannot easily be expressed using NL. Also, the expert may have become so accomplished at his/her task that its performance is now 'automatic', done almost without thinking about it, and, as a consequence, the knowledge required cannot easily be retrieved and called to mind. When the expert is unable to provide the actual knowledge used, then she/he may say nothing, or else give an ostensibly plausible 'textbook' explanation.

2.3.2 Practical Limitations

There are a number of practical limitations that blight manual KA from NL. One of these is the time that the process takes; the interviews and protocols themselves can take a long time, but the subsequent transcription and analysis can take many hours or even days to complete. When one considers the amount of preparation that may be necessary, and the fact that it is likely that a number of sessions will be needed (and sometimes a session will fail to achieve its aims), then the time taken begins to represent a serious obstacle in the system development process.

Other problems surround the chosen experts. The length of time that is typically needed for KA and the high value that is placed upon experts' time means that gaining access to them can be difficult. Even supposing that their knowledge is expressible, they may not give it to the knowledge engineer if they are inarticulate, if they perceive a threat to their jobs from the new technology, or if they are embarrassed to reveal their actual working methods. They may not even be real experts at all, but just happen to be the people who are performing the task within the organisation.

Finally, problems can arise due to the knowledge engineers themselves. The methods discussed above depend to a great amount on the abilities and experience of the knowledge engineer. An inadequate grasp of the domain in hand, or of KA or knowledge representation techniques, the misinterpretation of or a lack of rapport with the expert, or a misunderstanding of the aims of the current KA process can all serve to undermine the acquisition sessions. (And, as Winston (1993) observes, knowledge engineering remains very much "an art, and some people become more skilled at it than do others.")

Together, these limitations can cause the 'knowledge bottleneck' in intelligent system development. Despite these problems, however, it should be emphasised that successful intelligent systems have been constructed by acquiring knowledge from NL in this manner, and NL represents probably the best source of and medium for communicating knowledge. The following section describes a number of techniques that have been developed in an attempt to overcome, by automating the task to some degree, some of the more practical limitations of acquiring knowledge from NL.

3 Automatic and Semi-Automatic Knowledge Acquisition from Natural Language

This section describes some of the attempts that have been made to automate (or at least, semi-automate) the process of KA from NL. The explicit goal of their authors has been to facilitate the acquisition of knowledge from the rich, but problematic, source that is NL. Ideally, this would be done using an NL understanding system, which would be able to comprehend its input in a human-like fashion, and so, acquire knowledge from it. However, the current level of sophistication of such systems (see section 4.7 for a more detailed discussion of the state of the art) has meant that researchers have turned to more expedient approaches, which often promise a decrease in the amount of time spent on KA, rather than an increase in the quality of the knowledge acquired.

Evaluating these attempts is difficult; it is hard to quantify how much knowledge *should* be learned, given a particular text in a certain context. Subsequently, few researchers are able to supply meaningful figures that convey some idea of the quality of their systems. The problems are compounded when trying to compare these approaches, given the wide variation in the assumptions made about the goals of the KA task, the nature of the input NL, the content and form of the background knowledge, the amount of human interaction required, and so on.

Hence, the purpose of this section is not to identify which are the 'better' approaches (and so, any accuracy values or other evaluation figures cited should be treated with caution); rather, the intention is to try to convey something of the flavour of these approaches and the different perspectives of this KA task that different researchers possess. The manner in which these approaches are grouped together below and the titles given these groups are somewhat arbitrary and done more for convenience and, perhaps, to suggest some underlying research themes, than to imply any strong coherence in purpose or intention; indeed, an approach can often be placed into more than one of these categories.

The majority of the approaches described below attempt to acquire knowledge from textbook-type sources (usually in English), which (usually) have the advantage of expression in relevant, grammatical sentences with, in many cases, some sort of higher-level structuring of the text. However, before describing these approaches, two sections will summarise briefly attempts that have been made to automate KA through interviews and through PA.

3.1 Automating the Interview Process

Recognising that the human expert remains perhaps the most important source of problem-solving knowledge, and that the interview is perhaps the most natural way in which to try to access this knowledge, there have been a number of attempts made to automate, at least to some extent, the interview process for KA (e.g., the TEIRESIAS (Davis and Lenat, 1982), MOLE (Eshelman et al., 1988), SALT (Marcus, 1988) and AQUINAS (Boose and Bradshaw, 1988) systems, and, more recently, that of Blythe and Ramachandran (1999)). Primarily, these attempts have been aimed at developing, or maintaining, an existing, partial, and perhaps incorrect, knowledge base; the interview system is integrated, to a certain extent, with the knowledge base and, in this context, will prompt the expert to suggest, for example, an addition to the knowledge to remedy the generation of an incorrect solution.

The ostensible strengths of this approach are that the knowledge is entered directly into the system by the expert without the need for a human intermediary, thus reducing the knowledge engineering effort (but making greater demands of the expert – see below) and that the interview system can make use of the existing knowledge base and knowledge of the appropriate problem-solving methods to provide a context and focus for its questions.

However, the dialogue generated by these systems is not very sophisticated, generally being produced through the instantiation of question templates with the internal names of domain concepts, and restricting the number of responses available to the expert; hence, it cannot really be considered to be a natural language dialogue. Typically, such an approach will require the fundamental structure of the knowledge base and the underlying problem-solving strategies to be static, so that any alterations to the knowledge can readily be accommodated. This technique also makes a number of demands upon the expert; McGraw and Harbison-Briggs (1989) identify certain criteria that the expert must meet if this approach to KA is to succeed:

- The expert should be cognizant of the problem and the way in which he/she would approach its solution.
- The expert should be able to conceptualise about the domain.
- The expert should be able to analyse his/her own knowledge.
- The expert must be motivated to use the tool in a conscientious manner.
- The expert should be able to assure the performance of the model that he/she encodes.

To which might be added the fact that the expert must be available as needed during the development and maintenance phases; since this could take quite some time, this requirement might represent too great a commitment of the expert's time. Nonetheless, this represents a practical approach to building intelligent systems, one which recognises that any knowledge engineering exercise is likely to result in partial and faulty knowledge to some extent, and that the knowledge base needs to evolve if the system is to continue to be useful over any extended period of time.

A similar approach which may help to overcome, to a certain extent, the problems that the expert can face when trying to verbalise knowledge is provided by the *Ripple Down Rules* (RDR) technique (Compton et al., 1989). This approach is founded on the realisation that, while not always able to explain how they reached some conclusion, in a particular context (for example, talking to a student, or to a knowledge engineer), they are often able to justify the conclusion –

and, moreover, that this justification alters depending upon the context. The basic idea is to include the modifications that the expert provides as a new rule within a nested IF…ELSE IF…ELSE IF…. structure; this rule will only be fired in the context provided by the firing of the foregoing rules.

3.2 Automating the Protocol Analysis Process

There are a number of tools available to assist the knowledge engineer with the analysis of protocols. For example, in the Keats KA system (Motta et al., 1990), the Acquist hypertext-based tool (Motta et al., 1988) is available to assist the user during data analysis. This allows text (which may come from PAs, or interviews or textbooks) to be split into fragments. These fragments are then attached to concept labels, and concepts can then be formed into hierarchies, with the user defining relationships to link concepts. The commercial *PC-PACK* tool provides a working environment in which the transcript can be annotated, edited, and highlighted with markers of different colours to indicated knowledge 'chunks' of different types.[3]

The KRITON tool (Diederich, Ruhmann and May, 1988) attempts to go further and actually automate some of the PA procedure. The transcription is automatically partitioned into sections (delimited by the expert's pauses), and each section undergoes a semantic analysis in order to identify propositions in the text. The appropriateness of the selected operators and arguments is then checked (by a human), and an attempt is made to instantiate any variables using the current contents of the domain knowledge base.[4]

3.3 Text Analysis Approaches

The TOPKAT system (Kingston, 1994) adopts some techniques from NL processing with the expressly purpose of aiding KA from NL. A lexical tagger is used to identify nouns in the transcription of an interview with a domain expert (presumably, the approach would work for textbook passages as well). Those nouns that appear in this text with a frequency greater than would be expected in everyday speech are tentatively identified as domain concepts. These are then presented to a human, who weeds out any irrelevant nouns. The presence of an adjective immediately preceding one of the identified concepts suggests a value of some attribute of the concept - a human is asked to supply some name for this attribute. While this system is performing a rather basic analysis, and one which requires human intervention, the advantage of such an approach would seem to lie in its offering a significant reduction in the time and effort required for this type of analysis.

A somewhat similar approach is adopted for the acquisition of background domain knowledge from textbooks in both the KRITON (Diederich, Ruhmann and May, 1988) and KITTEN (Shaw and Gaines, 1988) systems. Words occurring with greater than usual frequency in the text are assumed to represent domain concepts. These words then used, at least in the case of KITTEN, as the basis for further knowledge acquisition episodes with an expert.

---

[3] PC-PACK *was* available from Integral Solutions Ltd.; since this company has been taken over, however, it is not clear if the package is still available.

[4] The article that describes KRITON does not make clear the extent to which this procedure for PA has been automated, and the extent to which it represents the authors' ideas for future development of their system: their approach and the extent of the automation which they suggest do seem rather optimistic.

3.4 Ontology-based Acquisition

Many of the techniques for automatic KA from NL make some use of ontological knowledge of the target domain to provide background knowledge for the acquisition process: the use of an ontology seems an appropriate way to incorporate semantics into a system, and provide a link between the syntax of the input and the semantics of the knowledge. In recent years, a number of general ontologies have been made available, and while their use has the potential of reducing the amount of effort needed to build the system, many researchers seem content to construct and use their own.

One approach that both uses ontologies and has as its acquisition goal the extension of those ontologies is presented in (Hahn, Klenner and Schnattinger, 1996a, 1996b; Hahn and Schnattinger, 1997; Hahn and Schnattinger, 1998). This approach is based on exploiting a combination of linguistic patterns and ontological knowledge. The given text, assumed to be concerned with a particular domain, is parsed. To try to incorporate potential new concepts into the ontology, information from the parse is exploited. For example, the combination "…operating system OS2…" suggests that the unknown quantity "OS2" is an instance of the concept "operating system", whereas "…a computer with OS2…" suggests that "OS2" is some constituent of a computer system. In the case of the latter, if the partial ontology lists a number of concepts which fulfil the role of computer system constituent ("monitor", "hard drive", "operating system"…) then, for each of these constituent concepts, a hypothesis might be formed to the effect that "OS2" is of that type. To reduce the number of hypotheses other linguistic and ontological evidence is used, if possible; in the case of the latter, this is in the form of the attributes and role information inherited by each of the hypothetical concept types. If more than one hypothesis remains after this process, however, it will be necessary to ask a human operator to resolve the issue.

The authors describe their evaluation of a system with an initial knowledge base consisting of 325 concept definitions and 447 conceptual relations, and operating on a total of 101 texts drawn from computer magazines. Under the most favourable conditions, this system is able to operate with a recall rate of 98% (which reflects the degree to which the target concept is in the returned hypothesis set), a precision rate of 39% (the ratio of correct to incorrect hypotheses) and a parsimony rate of 31% (the ratio at which the returned result is a single, correct hypothesis).

3.5 Pattern- and Template-Matching

Another theme common to many approaches is the idea of matching the NL input against some preconceived notion of the knowledge expressed in it. Obviously, the understanding of language involves identifying patterns at some level; for automatic KA systems, however, the patterns used are often at a much coarser level of granularity, and explicitly incorporate assumptions about the nature of the NL input, and about the representation and content of knowledge of the domain and of the target knowledge to be acquired.

The PETRARCA system (Verladi, Pazienza and Magrini, 1989) represents an attempt to acquire knowledge of word definitions from a corpus of NL texts. In this case, word definitions are in the form of lists of *surface semantic patterns* (SSPs), effectively cases in which the words are used (the authors argue that a definition of this sort is both a natural way for defining a word and one which may be acquired inductively from a body of examples). The corpus consists of press agency releases concerning finance and economics. The text is subjected to morphological and syntactical analysis to identify words and the roles that they play in sentences. Next, the system attempts to derive semantic interpretations of examples of unknown (noun?) words in the text. It uses elements of general and "in part, generic"

knowledge to do this: *syntax to semantics* rules (simple patterns which match words to semantic concepts associating them - for example, the word "of" between two words suggests that they might be related through the concept of 'possession') are used to generate possible interpretations; *conceptual relation* rules, which describe the semantic constraints on the associations, are then applied to try to remove unlikely semantic interpretations, with the aid of a *type hierarchy* (to identify in more general terms the roles that words are playing). The remaining associations form the basis of new SSPs (following an attempt to generalise the results of this analysis, again using the type hierarchy, and, perhaps, with some operator intervention). The extent to which knowledge of this sort is useful is not, however, immediately apparent, and the background knowledge required to perform the acquisition seems to require a certain amount of effort to obtain.

The work of Hahn, Klenner and Schnattinger described above also uses very general linguistic patterns, as, in their case, cues for hypothesising extensions to their ontology. In a somewhat similar vein, Bowden, Halstead and Rose (1996) describe an approach to extracting particular types of general relational knowledge from (non-domain specific) texts. Their technique involves identifying *triggers* in the text for particular types of relation; for example, for definitional knowledge, a trigger phrase might be "…is defined as…." (There are also 'negative' triggers – "…cannot be defined as…"). For each type of relation (the authors consider three types, namely definition, exemplification and partition) there is a set of positive and a set of negative triggers, initialised by a manual analysis of texts and developed through a process of trial-and-error, each containing around 20 triggers. Using these sets, potential knowledge-bearing phrases are identified in a pattern-matching fashion, and the knowledge extracted. An intermediate part-of-speech tagging stage can help to improve the results. The reported results, on a limited test set, suggest that this approach is able to achieve 100% precision for all relation types, with varying recall rates (e.g. 33% for definitional knowledge). Shortcomings of this approach include a high number of false triggerings (not reflected in the given results), the need for a 'chunk' of knowledge to be contained wholly within a single sentence, the problems of conflicting triggers, the limited handling of anaphora, and a lack of robustness to variations in punctuation, all of which might be ascribed to the simplicity of a pattern-matching approach to NL texts.

Patterns that are slightly more specific to the nature of the text input are used by Moulin and Rousseau (1992) within their SADC system, with which they attempt to acquire knowledge through the analysis of only the logical structure of a text, and not its semantic content. In common with a number of researchers, they choose to acquire this knowledge from technical texts (in particular, building regulations), in which a certain level of formality and structure can be expected. Their approach relies to a great extent on an analysis of the nature of such texts, which is reflected in the grammar that is used by SADC. For example, they identify that key words such as "must" and "cannot" in some sentence (potentially) indicate the presence of some regulatory rule in that sentence, and words like "unless" and "however" may indicate exceptions to a rule.

Using this grammar, a partial parse of the text is performed in two passes. In the first, potential rules are identified, along with suggestions for the domain concept to which the rule refers, and the conditions and conclusions of the rule. A second pass, at a higher level identifies meta-textual statements (e.g. "the following section applies in the case of…") that must be used to alter the conditions of any rules to which they refer. Throughout this process, a human domain expert is required to edit the suggested rules, selecting appropriate domain concepts, and adding new ones into the domain hierarchy, to resolve anaphora and other semantic difficulties. A knowledge engineer might also be required to modify the grammar rules in the event of a parsing failure; Moulin and Rousseau claim that the grammar becomes

near-stable, after a period of development (with, in their case) around 400 grammar rules. The extent to which this grammar is applicable to similar texts in other domains is not clear.

In this fashion, the SADC system can help to make the logical content of the text explicit (and can serve as a check on the consistency of the original document). To convert this into a rule base for an expert system would, the authors admit, probably require some form of semantic analysis of the text (to decide upon, for example, the appropriate values that a concept might assume). Nonetheless, their approach describes a practical approach to KA.

In contrast to Moulin's and Rousseau's work, the system developed by Virkar and Roach (1988) purposely exploits semantic knowledge of texts and their domain. Their research addresses the problem of maintaining the knowledge within an intelligent system current in the face of continuing scientific and technological development of the field; in their field of interest (pharmacology) the rate at which new knowledge is being discovered is such that any intelligent system in this area would need to be revised at frequent intervals if it were not to become obsolete. The manner in which Virkar and Roach propose to do this (for their expert system for predicting drug interactions) is by examining the abstracts of published research papers, and incorporating the knowledge that can be gleaned from these into the knowledge base.

The text of an abstract is subjected to several levels of analysis. A *transformational semantic grammar* (containing about 20 transformations) isolates parts of sentences that contain a single 'knowledge concept'; it performs this decomposition by locating *function words* (such as "and", "or" and prepositions) within the text. A *semantic grammar* (containing more than 40 patterns) is used to parse these sentence fragments. The grammar relates how certain cues, to be found within these fragments, are associated with particular domain knowledge structures, which, in this case, are frames. For example, the pattern "increase…concentration" is associated with a 'drug interaction' frame, and when encountered is the signal to create a new frame of this type. This frame is then instantiated as completely as possible using both information from the abstract and the contents of the domain knowledge base.

Finally, a set of *text grammar* rules is used to connect the various frames into a coherent whole. Grammars of this type describe, at a high level, forms which the text might be expected to take (e.g. the abstract might begin with some hypothesis then describe the experimentation used to test this hypothesis, the measurements taken, and the conclusions drawn). Using patterns of this form, then, it is possible to link the disparate frames into a coherent 'story'. This linked collection of frames can then, in theory at least, be assimilated into the knowledge base (however, for a number of reasons (such as maintaining the consistency of the original knowledge base) this is not done in practice; instead the structures are stored in a subsidiary knowledge base, which is used by the expert system if the principal knowledge base is found lacking).

The grammars used in this approach are, to differing extents, predicated upon the particular manner of expressing knowledge in the pharmacology domain, and the relationship that this mode of expression has to the sort of knowledge that the system wishes to acquire. New domains would need new grammars, requiring extensive analysis of the source texts. However, this effort might prove worthwhile if it were to prolong significantly the effective operational life of a system, when one considers the cost of developing expert systems from new.

The Wit system (Reimer, 1990a; 1990b) attempts to acquire knowledge of concepts from technical texts using a limited parsing mechanism that recognises only a small number of "syntactical phenomena". For example, the composite noun pattern $n_1 n_2 ... n_m$ (where n_i is a noun and n_m corresponds to some known concept) suggests a new concept, subordinate

to the known one in the domain type hierarchy. Meta-level *textual coherence* patterns are used to determine the 'scope' of the description. Words that do not match any of the patterns are ignored. Initially, the system is provided with a high-level domain ontology (of, in the given example, computer-related devices), which is used to identify those concepts of interest in the text. The KA takes the form of extensions to this ontology, and, as such, is similar to the work of Hahn, Klenner and Schnattinger described above. In addition, the system incorporates an inductive learning element, by which the descriptions of concepts can be revised and generalised in the light of newly acquired knowledge.

Hull and Gomez (1998) present a system for automatically acquiring knowledge from biographical entries in an encyclopaedia (this work represents an extension of their earlier SNOWY system (Gomez et al., 1994; Gomez 1995), used to acquire knowledge from encyclopaedias about the diet and habitat of animals). Entries are (partially) parsed, using a lexicon of 75000 words (parse success rate of 75%) and then passed to a semantic interpreter, which instantiates *verbal concepts*, frame-like structures associated with particular actions, filling slots with the appropriate subjects and objects. Basic inferences are made to provide some of the details. Additional knowledge is provided in the form of a list of verbs which are 'interesting' (determined by an analysis of the content of a sample set of biographies, and selecting those that occur with the highest frequency, aside from the most common verbs such as "to be" and "to have") along with an indication of the verbal concepts that these relate to, and a general ontology, which helps to disambiguate phrases and complete the instantiated frames. A set of around 90 question templates, completed with the name of some historical figure, serve to define the sort of questions that can be asked of the system.

In a comparative test, it was able to answer questions with a precision of 85% and a recall of 54% when compared to the sample answers produced by humans (70% of the answers supplied by the humans were returned by the system). Furthermore, the system completed its answers in a fraction of the 12-15 hours it took the humans to complete the task. It might be argued that the task was eased somewhat by the nature of encyclopaedia entries, which tend to be concise and clear, and information-rich, but, nonetheless, the reported performance of this system is impressive.

3.6 Memory-Based Acquisition

Lebowitz' Researcher system (Lebowitz, 1983a; 1988) represents an attempt to acquire knowledge of concepts from text by integrating 'bottom-up' syntactic processing of text and 'top-down' re-use of similar domain concepts stored in the system's memory. Assuming a basic initial memory of the system, the task of 'understanding' text describing a new concept becomes one of recognising which concepts in memory are most similar to it, and the ways in which it differs from these. The working domain of this system is that of disk drives, and the texts to be processed are patent descriptions of these artefacts. The goal is to acquire conceptual descriptions of the physical structure of the disk drives. A frame-based model has been developed for describing the drives and their constituent components, along with a set of relations that describe the physical associations of these components.

The text is processed and tagged in such a way as to identify "memory pointers", words, usually noun phrases, that refer directly to objects in the memory, and also "relation words" which describe how the various concepts in the text are associated (hence the technique involves a degree of pattern-matching). Heuristics are used to cope with prepositional phrases, to decide upon the focus of the text, etc. On the basis of the memory pointers, new instances of existing concepts are created and then instantiated using information from the text. Finally, the system attempts to perform clustering on new and existing concepts, modifying the descriptions of higher-level concepts in order to refine

its (generalisation-based) memory structures in the light of the new knowledge acquired. The knowledge of the system can be probed during a NL question-answering phase, which makes use of a similar method of "memory pointers" from the questions to attempt to locate where the answers might lie in memory. (Lebowitz' IPP system (1983b) is similar to the Researcher system; however, instead of operating upon patent texts and reasoning about physical devices, it deals with news reports and events. Consequently, the knowledge and memory structures in IPP reflect this difference.)

This drawback of this approach would seem to be the extent to which it is domain specific. The cue words refer to the devices in question, and the memory structures (and the memory itself) are tailored towards the particular domain. Furthermore, the text of patent descriptions (and news reports) seems to be quite concise and focused, and it is not clear how well the technique would extend to less amenable text.

A similar approach is to be found in the KA system (Goel et al., 1996), albeit with more sophisticated notions of how devices might be stored in memory. The aim of this system is acquire *structure-behaviour-function* models of devices. It is supplied with an ontology of the concepts used to describe such models, along with an existing memory of devices described in this fashion. The input to the system is the description of some new device (taken from a book entitled "The Way Things Work"); this description comprises an explanatory passage in English and an annotated diagram, translated into a symbolic representation. In a fashion akin to that of the Researcher system, the text is processed 'conceptually', using the domain ontology, to identify potential cues for the retrieval from the memory of the models of devices that are in some way similar to the new device. The text is also parsed, and using a semantic network and the ontology, is translated into a conceptual interpretation. This interpretation is used to try to identify functional and structural differences between the new device and the devices that have had their models retrieved. These differences are then used to select appropriate generic modification plans to enable a retrieved model to be altered so that it describes the new device; this new model can then be indexed and stored in memory.

3.7 Machine Learning

Several researchers have attempted to make use of existing machine learning algorithms to acquire knowledge; since these algorithms have been developed with the express purpose of capturing and representing knowledge, this would seem an attractive approach to adopt. In general, however, such algorithms expect their input to be presented in terms of constrained sets of attributes, so they cannot be applied directly to NL; it is necessary to process the NL into some form that is acceptable to the chosen algorithm.

Inductive clustering algorithms are used in the Researcher and Wit systems, both described above, for refining the acquired knowledge and situating it appropriately in the system memory.

Maedche and Staab (2000a, 2000b) suggest a method for machine-learning non-taxonomic relationships among concepts. A text is, by turns, tokenised, matched against a lexicon, subjected to lexical analysis and parsed. Using a set of heuristics and a domain ontology, the processed text is then further analysed to try to form pairs of general concepts which are related linguistically within the text and which may be related conceptually.

These concept pairs are then supplied to an associative learning algorithm; this algorithm attempts to identify those concepts that consistently co-occur with a frequency that suggests that there is some form of relationship between the two. As might be expected, an evaluation of the outcome of this approach proves to be quite difficult, and the authors introduce a rather complicated testing strategy. The given results suggest that the algorithm can find little *support* (the percentage of examples that contain the two concepts) and little *confidence* (the percentage of examples containing one

of the two concepts that also contain the other) for the relationships in this manner. However, this might well be due to the nature of the learning algorithm itself, and this approach remains an interesting one.

Faure and Nédellec (1999) present a method for learning both ontological knowledge of domain concepts and general verb 'subcategorisation frames' (SFs), by considering nouns that play the same syntactic role relative to the same verb to be related semantically. Technical texts from some particular domain are parsed, and then head nouns and prepositional phrases are attached to their associated verbs to form specific SFs. From these SFs, nouns are considered together to form a *basic class* if they share at least two different SF contexts. A clustering algorithm is then applied to these basic classes to form successively more general concepts; human intervention is required at this point to validate the concepts formed, and to prevent over-generalisation. The general concepts learned can then, if appropriate, be used to generalise the SF descriptions. (de Chalendar and Grau (2000) describe a similar approach to acquiring semantic knowledge of words, without the learning element (so no attempt is made to cluster classes of words into higher-level concepts), but without the restriction to a single specialised domain, and with less reliance on human intervention. Since the raw input to their system consists of news reports concerning dealing with quite diverse topics, an initial phase attempts to collate portions of texts into *thematic units*, based on the common usage of words, and the degree of *cohesion* (based on co-occurrence) of these words. Similar thematic units together form a *semantic domain*; the system can then try to identify the basic classes of nouns that exist within a particular semantic domain, in a fashion akin to that proposed by Faure and Nédellec.)

Delannoy et al. (1993) outline an approach to extracting rules from technical texts. These texts, which are relatively clear and unambiguous, consist of explanatory passages interleaved with examples. Their idea is to parse the text into Horn clauses, and then apply Explanation-Based Learning with knowledge abstraction to derive general rules. At the time of writing their paper, the system had yet to be fully implemented, so it is difficult to decide on the merits of their approach.

An approach to assist the knowledge engineer who is presented with the task of extracting knowledge from a large set of texts, in which sections relevant to particular areas of knowledge might be widely dispersed, is outlined in (Lapalut, 1996a; 1996b). The texts are analysed morphologically to identify words that are 'meaningful' – which, in practice, are considered to be the verbs, nouns and adjectives. The texts are then partitioned into segments, each of which contains some fixed number of these meaningful words. A matrix is then formed, with the meaningful words as rows and the segments as columns; a '1' at some position in this matrix indicates that the corresponding word appears in the corresponding segment. This matrix represents the input to a conceptual clustering algorithm, which forms a hierarchical classification of each of the rows. This classification can then be used to provide some indication of related sections in the texts.

The accuracy and usefulness of this technique cannot be made from the presented results; however, it makes relatively few demands in terms of a priori knowledge of the domain and the nature of the texts, and might provide guidance to a knowledge engineer faced with correlating a large number of texts.

3.8 Interactive Approaches

Szpakowicz (1990) presents a semi-automatic approach to the capture of domain knowledge from technical texts, by which a conceptual model of the domain is constructed incrementally. He makes a number of assumptions about the task: first, that the domain does not need expertise to be understood fully (and so, the domain knowledge can be

expressed verbally in its entirety); secondly, that the text describes the domain fully (presumably for the KA exercise to be considered completed after processing it); and, finally, that a skeletal conceptual network is available at the outset of processing (in the cited example, this network contains 300 concepts, describing activities, objects and abstractions).

When processing the text, within 'meaningful' sentences (the notion of 'meaningful' is left vague; presumably the human operator determines what is meaningful and what is not), verbs are associated with activities and noun phrases with objects playing various roles in activities ('meaningless' sentences in the text (such as examples) are ignored). The operator adds any unknown words to the lexicon (which, initially, contains around 300 words), and there is no attempt made to resolve automatically anaphora or idiom. In the event of an incomplete analysis, for whatever reason, the operator is asked to rephrase the text in a semantically equivalent but more comprehensible (to the system) manner.

A semantic mapping phase then translates the identified items into a conceptual representation, with unassigned parts of the sentence referred to the operator. If the fragment of network produced by this mapping can be integrated within the main conceptual network, a matching routine is invoked to try to determine the areas of this network that may be associated with this new fragment. If there is sufficient 'overlap', the fragment can be incorporated; if not, the operator is prompted to supply some association between it and the concepts in the network.

In this manner, nouns, verbs, adjectives and adverbs can all be incorporated within the conceptual network. It is difficult to judge the success of such an approach, and the amount of user interaction.

The DASERT system of Biébow and Szulman (1993) attempts to check automatically the consistency of functional specifications (expressed using a formal modelling language); these specifications contain NL comments which can be used during this checking process, and to extend the system's knowledge of the specification. In this way, DASERT may be considered to be performing KA. In a manner similar to that adopted by Szpakowicz, the text is analysed lexically, linking words with concepts in the knowledge base, syntactically, establishing the relationships between words, and then semantically, with the construction of interpretations of the relationships in the form of semantic networks. After every step, the operator is asked to confirm or correct the results, and in so doing, extend the lexicon, incorporate new concepts into the hierarchy, and so on.

The TERMINAE system (Biébow and Szulman, 1999), which aims to provide an environment in which knowledge engineers can develop domain ontologies, incorporates mechanisms for analysing texts and suggesting possible domain concepts, and then for maintaining associations between a new concept and its occurrences in the texts (as a part of the definition of the concept), and proposing generic associations between this concept and existing ones, in order to facilitate its inclusion into the ontology.

Lu and Cao (1990) present a somewhat idiosyncratic semi-automated approach to KA from NL in the form of textbooks. Since comprehensive automatic NL understanding is not yet a feasible approach, they suggest that KA might be achieved by initially translating texts by hand into an intermediate, unambiguous and constrained formal language, which can then be processed by a computer, and the contained knowledge extracted. The authors have developed such a formal language (for Chinese texts, so it is difficult to draw any conclusions using the examples they provide), but this approach does not seem to be particularly practicable for a number of reasons, not least of which is the time taken to develop this formal language (if, indeed, it is possible to do so without losing information and accuracy from the text) and the time taken to translate textbooks manually (which, in any case, might be tantamount to performing KA on the text).

3.9 Integrated Approaches

Mikheev and Finch (1995) outline a Knowledge Acquisition Workbench for extracting domain knowledge. This uses a number of different techniques, similar to some described above, for semi-automatically identifying and clustering terminological knowledge. A major consideration in their work, however, is that the incorporation of new and alternative modules for performing sub-tasks during this analysis should be facilitated as much as possible, as the exploitation of more general existing language resources (such as thesauri and ontologies). They attempt to achieve this by insisting upon a common protocol (based on SGML) to govern the data-flow between modules.

In a similar vein, Aussenac-Gilles, Biébow and Szulman (2000), noting the lack of maturity of tools for automatic KA from NL and, as a result, the limited use that they find outside academic research, suggest that one approach to the acquisition of knowledge – specifically, in their case, of an ontological form – from texts is to bring together a number of these tools within a common framework. Human knowledge engineers would then be able to select the appropriate tool(s) for performing (or, at least, assisting to perform) the KA sub-tasks that arise. In so doing, the aim is to be able to exploit the particular strengths of a tool, with the availability of alternative tools compensating for its weaknesses.

3.10 Text Data Mining

In addition to the work on automatic KA, there has been much interest in the topic of *text data mining*. The distinction between this and KA is not always an easy one to draw; however, where KA is concerned with the extraction and representation of a body of knowledge that can then be applied to perform some specific intelligent task, text data mining usually involves the (automatic) inference of some novel scrap of knowledge through the analysis of some given set of texts. The usefulness of such techniques often lies in their ability to consider data in greater numbers than is possible for a human.

An interesting example of this may be found in the work of Swanson and Smalheiser (1997). Through the (semi-) automatic analysis of the titles of scientific papers in the biological and medical fields, they have been able to postulate a previously unknown association between migraines and magnesium deficiency. Put simply, their approach relies on identifying a set of articles which report an interesting association between variables A and B and a different set of articles which report some association between variables B and C, and there are no articles concerning a link between A and C, then this is a potentially rewarding association to investigate (and potentially new knowledge). Since the number of associations found in this manner could be extremely large, it relies on some sort of intervention to guide the process in a useful manner, and also upon the nature of paper titles in the examined field. Whether such an approach can find a wider application remains unclear.

However, the emphasis on discovering *new* knowledge, rather than trying to acquire existing knowledge effectively means that further consideration of text data mining is outside the scope of this survey, and it is mentioned here for the sake of completeness.

3.11 Summary

This section has described a number of methods and techniques that have been developed to automate some or all of the task of KA from NL. It is difficult to assess the scope and success of these approaches, given that the aims of their authors and the assumptions that they make about the task, vary considerably. However, it seems clear that the field is still relatively immature (and the difficulties of appraising the techniques may be a reflection of this), with few, if any,

of the systems described above currently being used during actual KA exercises. Nonetheless, it is possible to highlight a number of common issues that the approaches raise, which can be summarised as follows:

- What types of knowledge can be expressed through NL? How might the NL inputs that contain a particular type of knowledge be characterised? Which types of knowledge is it practical to try to acquire from NL, and which can be captured more effectively from other sources? (Many of the above approaches focus on the acquisition of ontologies/hierarchies of concepts, which suggests that there is some shared belief that this sort of knowledge is expressed through NL, and can be acquired from it (alternatively, of course, the concentration of work in this area may be merely a reflection of the comparative value or scarcity of this type of knowledge).)
- The source and nature of the NL: if the input is to come from, say, textbooks, then it might be possible to make useful assumptions about the form and the content of this knowledge, and, for example, to exploit the use of headings within the text to demark 'knowledge boundaries'. If, on the other hand, the input is to be in the form of the transcription of an interview with a human expert, then different assumptions will hold (researchers have thus far shied away from attempting unconstrained, direct automated interviews with experts).
- The acquisition 'algorithm' itself. What is the underlying basis for asserting that this process is identifying and extracting knowledge from NL? Is the 'knowledge' that is acquired expressed in a useful form?
- The degree of human intervention or interaction that is necessary. If it is assumed that a human operator is to be involved in the process, then it is possible to devise a useful partitioning of the workload into the more repetitive and time-consuming tasks that the computer is able to do and the more difficult tasks that the operator can perform. However, consideration must be given to the abilities of the operator, and to whether or not this interaction represents a constructive use of her/his time.
- The amount, nature and source of the background domain knowledge that is necessary. Many of the approaches described above make use of domain knowledge of some form or other to provide some sort of semantic context for the KA process. As is always the case for intelligent systems, consideration must be given to the question of whether it is feasible to assume that this knowledge is available, and in the form required. And assuming that it is available, what is its source, and how is it to be acquired?
- The nature and extent of the 'parsing' knowledge in the system. 'General' complete NL parsers are difficult to construct and, once built, tend to be fragile and to produce many alternative parses for apparently simple sentences. Aside from the practical difficulties, a complete parse may not be necessary for useful KA and for these reasons most of the systems described above aim at only a partial parse of their input. This may be done in a domain-independent and 'source-independent' fashion, but is often done by exploiting what is known of the way that target knowledge is expressed in the domain within the type of NL input selected for processing. The use of patterns of this sort can provide a useful (and perhaps necessary) link to the semantic representations of the knowledge, but their use will restrict the generality of the system, and perhaps of the approach, if similar patterns are not found to exist in other domains. Furthermore, the development of these patterns would seem to be a complex KA exercise in its own right, and one that is often done in an ad hoc fashion.
- A related issue is the extent to which is possible - and, indeed, desirable - to reuse existing language resources in KA systems; these resources would seem to offer a quick and convenient means of introducing additional

knowledge into systems, but may introduce unnecessary and, perhaps, detrimental complexity and assumptions into the system. If such resources are to be exploited, what are the appropriate architectures for doing so? (The reuse of language resources to reduce the time and effort involved in constructing systems) is a major current theme in language engineering. The topic of reuse is discussed further in section 4 and, in particular, in appendix A below.)

- The use of other learning techniques. Several of the systems described above make use of explicit ML algorithms to acquire the knowledge. While this is an attractive combination of different areas of AI research, the relative lack of sophistication of current algorithms of this sort can means that an unreasonable amount of processing of the NL input is required and the quality of their learning falls well below that which might be expected of humans. It is probably for these reasons that the investigation of such hybrid approaches remains relatively limited.
- The usefulness of the adopted approach. If the form and content of the knowledge that is acquired is inappropriate or if it can be acquired more simply from elsewhere, then this calls into question the whole exercise. Similarly, since their value is often considered to lie solely in the increased efficiency that they confer on the KA process, the amount of preparation of the system, and of tailoring to a particular domain or task that is necessary may mean that this efficiency is gained at the cost of increased time spent on system development – which may not represent the most effective allocation of resources.
- The evaluation of systems. As mentioned above, it is difficult to perform evaluation of the systems, and even more difficult to make a quantitative comparison of several systems. However, some sort of comparative evaluation of this sort would seem to be necessary if research in the field is to progress and if, eventually, the systems are to be used in real applications. While the techniques applied remain, to some extent, expedients in lieu of complete NL understanding, some measure of their performance in the context of the development of practical intelligent systems, when compared with that of the more conventional approaches to KA currently used, would appear to be needed.

These issues represent major research questions that must be addressed if the field is to progress towards maturity.

4 Language Engineering and Knowledge Acquisition

Language Engineering (LE) is the "discipline or act of engineering software systems that perform tasks involving processing human language" (Cunningham, 2000). There are a number of different areas of current research in LE that could conceivably represent KA from NL or otherwise assist in this process. Accordingly, this section will introduce and describe the following areas of LE, outlining the main implementational approaches to each and trying to provide some indication of the state of the art:

- Information Extraction
- Information Retrieval
- Machine Translation
- Speech Recognition
- NL Generation
- NL Understanding

There would seem to be two distinct, general approaches to constructing (the components of) systems to address these tasks. The first — which might be called the 'knowledge engineering' approach — involves building the knowledge structures (grammars, parsers, etc.) manually, relying on human experience of the field and the identification and encoding of useful heuristics. The second approach — the 'statistical' approach — involves the use of statistical models derived automatically (or, at least, semi-automatically) from large corpora of relevant, annotated textual material. Each approach has its own particular strengths and weaknesses: the knowledge engineering approach promotes a greater understanding of the problem but can require a great deal of human expertise, while the statistical approach is seemingly better suited to the task of implementing systems able to address large-scale, real-world problems, but requires processed data in large numbers. Whatever the particular approach adopted, in recent years an increased emphasis has been placed upon the reuse of existing resources during the development of LE systems; given the time and effort involved, effective reuse seems to be the most realistic approach to the construction of large systems. With this in mind, appendix A describes briefly the types of resource that are currently available.

While a general NL Understanding system represents the ideal for KA from NL, the current state of the art is such as to render a system of this sort unfeasible. The current approaches to automatic KA described above have, in number of ways, a closer affinity with Information Extraction systems; accordingly, this area is dealt with at some length below, with the other areas being treated more summarily. As mentioned above, there appears to be no common notion of how KA tools might be evaluated. Since LE systems would seem to share certain characteristics with KA tools, it might prove useful, before commencing the overview proper, to devote a section to a brief discussion of the evaluation of LE systems. Furthermore, this should assist when attempting to convey the level of maturity of the LE techniques in the subsequent sections.

4.1 The Evaluation of Language Engineering Systems

In general, there are three kinds of evaluation that might be applied to a system (Hirschmann and Thompson, 1998):

- *adequacy evaluation* — the determination of the fitness of the system for some purpose.
- *diagnostic evaluation* — the determination (usually using some set of test inputs) of the performance of the system with respect to some clearly defined dimension of the possible inputs. (This sort of evaluation will mainly be of interest to the system developers, allowing them to assess the effect of changes to the system.)
- *performance evaluation* — a measurement of the system performance, usually made for comparative purposes.

While these measurements are not independent, they are different; for example, some system displaying relatively poor performance might yet be adequate for the particular needs of the user. In addition, it is sometimes possible, and desirable, to evaluate the constituent components of a system, both in their own right and as a part of the wider system.

In recent years, the emphasis has been placed upon performance evaluation (by, for instance, DARPA) in an attempt to focus attention upon the development of 'working' systems. This emphasis has had a number of beneficial effects — and also some detrimental ones. Hirschmann and Thompson summarise the advantages as being:

- Standardised test formats have been developed to allow comparisons to be made.
- This form of evaluation typically relies on the use of (large) text corpora and other resources — in order to standardise the tests, these resources have been developed, standardised (to a certain extent) and shared.

- Conferences and workshops have arisen around the common testing format, leading to an active and communicative research community.
- The availability of standard evaluation methods over a period of time has allowed the rate of progress of particular approaches to be observed, encouraging further research along those paths that seem most profitable.

The disadvantages they list can be summarised as follows:

- The focus on performance evaluation has tended to overshadow the development and use of adequacy evaluation methods; hence, it is difficult to judge the extent to which a system meets a stated need, and is usable in a practical context.
- Many of the evaluation methods are application-specific, leading all researchers to build systems that perform the same task, without necessarily developing the deeper understanding that will lead to methods that are transferable to other applications.
- A related problem is that there is little or no evaluation of the degree of portability of systems to new domains; the (real or perceived) high cost of transferring systems presents an obstacle to their adoption in practice.
- Evaluation is time-consuming; it can be difficult to achieve the right balance of testing and development.
- Excessive focus on evaluation can lead to the stagnation of research, with risky, but potentially extremely rewarding, approaches shunned in favour of tried-and-tested methods of incrementally improving system performance.
- There has been insufficient attention paid to the evaluation of systems operating with languages other than English.

In the following sections, the reported evaluations are, on the whole, those of the performance of the systems; when considering these values, one should bear in mind the limitations of this form of testing.

4.2 Information Extraction

The Information Extraction (IE) task is one of extracting relevant information from a natural language text, and presenting this information in some formalised manner. Hence, an IE system attempts to filter the content of a message, summarising the relevant information contained within it in an accurate and unambiguous manner. On the whole, IE systems do *not* attempt to process the entire document they are given. NL processing techniques for doing so are currently error-prone, brittle and rather slow, and the emphasis in the IE field lies firmly upon the production of fast, robust and practical systems. The goal of a particular IE task is presented as a short description of the kind of information that is sought (which may be in the form of a database template or schema specifying the output). One of the strengths of the field is that there is a widely accepted method of evaluation: an IE system's output is appraised according to its *precision* and its *recall* in performing some particular task:

- *precision* — a measure of the accuracy of the output produced:

$$precision = \frac{number\ of\ correct\ elements}{number\ of\ elements\ extracted}$$

- *recall* — a measure of the extent to which the total information contained in the text has been extracted:

$$recall = \frac{number\ of\ correct\ elements}{number\ of\ possible\ elements\ in\ text}$$

Since the late 1980s, IE systems have been compared at the regular Message Understanding Conferences (MUCs) held in the United States (MUC-6 was held in 1996 and MUC-7 in 1998). The comparisons are made based on some pre-selected task; competing system developers are supplied with a corpus of relevant texts and a specification of the required output, and their systems' responses are compared to those of human subjects performing the same task (which gives the 'ground truth' for appraising the systems). However, there is a degree of subjectivity in the exercise and the tasks are not always straightforward: some of the tasks are difficult to quantify, and human responses are often less than perfect.

There is often found to be some degree of trade-off between a system's precision and its recall; very precise systems often have low recall rates, whereas good recall is frequently accompanied by poor precision. The *F-measure* represents an attempt to combine precision and recall, giving a single value for the quality of the system:

$$F - measure = \frac{(b^2 + 1)PR}{b^2 P + R}$$

where P is the precision, R the recall, and $ß$ is a parameter representing the relative importance of P and R — a value of $ß$ greater than 1 indicating the greater importance of precision, a value less than one indicating that the emphasis lies upon recall (the F-measure represents an approximation to the geometric mean). (As will be seen below, IE systems are often constructed in a modular fashion; this offers the possibility of evaluating and comparing the constituent components in an incremental fashion.)

There is a certain amount of variation in the complexity of IE tasks; this variation arises as a result of a number of factors:

- the language in which the document is written;
- the style in which the document is written (for example, more formal documents may prove easier to process than do, say, informal emails);
- the nature of the content of the document — lengthy documents containing a great deal of irrelevant information can be difficult to process, as can documents including, and referring to non-textual information (figures etc.);
- the specification of the task itself — the depth and nature of the information sought, and the adequacy of the chosen internal structures for representing this information.

However, Appelt and Israel (1999) suggest that given the current state of the art, IE systems can be approximately 60% successful (that is, have an F-measure of 0.6). (Furthermore, these authors suggest that this figure may well represent something of a high-water mark in IE systems development using current techniques, since there seems to be some consensus that 60% equates to the typical amount of information that is clearly and explicitly stated in a document.)

4.2.1 The Architecture of Information Extraction Systems

Most researchers in the field view the IE task as a progression through a sequence of stages. These stages, in the general order in which they occur, can be summarised as follows:

1. Tokenization;
2. Morphological and lexical processing;
3. Syntactical analysis;
4. Coreference resolution;

5. Domain analysis.

It should be noted that not all IE systems implement all these stages — for instance, it is possible to stop after morphological and lexical processing and return to the user the information gathered up to that point. In general, the further 'downstream' a task is in this process, the greater the extent of its domain- and application-dependence. In the following sub-sections, a brief description of each of these stages is given, along with a discussion of some of the approaches that have been adopted to implement them, and, where available, some indication of the performance levels achievable by current technology.

Tokenization

This is the task of dividing the text into its constituent atomic elements (words, usually). For well-structured documents in English and similar languages, this is a fairly straightforward task, since words are separated by spaces and punctuation. However, for other languages (such as Chinese and Japanese) in which the boundaries between words are not marked by such typographical conventions, the task can be more difficult.

Morphological and Lexical Processing

Initially, this stage involves identifying the roots, prefixes and suffixes of the word tokens. Again, the difficulty of this task is somewhat dependent on the language used — English, for example, can be easier to process than a language displaying a greater degree of inflection. Next, some further analysis is performed to try to determine the function of these within the text — usually by a process of *part-of-speech tagging*, by which words are tagged as being nouns, verbs, adjectives etc., and *named entity recognition*, by which the actors in the event being described are identified:

- *Part-of-Speech Tagging*: Part-of-speech taggers can achieve good accuracy rates (around 95% correct) (Brill, 1994). They are usually developed statistically, either using supervised training (that is, using hand-tagged corpora of texts) or unsupervised training (the corpora contain no such tagging information — words are automatically clustered into tagged groups which are then used in to supervise the training). Hidden Markov Models (e.g. (Kupiec, 1992)) are commonly used for taggers. See (Merialdo, 1994) for an overview of stochastic approaches. There may be a need to develop a tagger that is tailored toward the working domains (and the typical uses of words in that domain), but 'off-the-shelf' taggers are available.
- *Named Entity Recognition*: Once this has been done, the crucial task of *named entity recognition* must be performed. This involves identifying the names of potentially relevant things within the document, and then 'typing' them — that is, determining the class (person, organisation, location, etc) of things to which the entity belongs. There are no rigid rules that can be applied to the formation of names, and it is not possible to have a look-up table of all possibilities. The knowledge engineering approach to building a name recogniser involves devising a lexicon (which may be domain dependent) of common names, and then, by a process of trial and error, developing a set of heuristics to cover words that are not in the lexicon but which seem to be assuming the role of names in the document. The statistical approach has centred on the application of Hidden Markov Models (HMMs) to this task. At MUC-6, the SRA system (Krupka, 1996), using a statistical component, achieved 96% recall and 97% precision (F-measure: 96.42) for the named entity recognition task: this is considered to represent a level of performance comparable to that of humans (it has been suggested that the performance for languages other than English might well be lower, which might merely be a reflection of the

inordinate amount of effort devoted to English language-based applications). Named entity recognition tools represent the most reliable form of IE system.

Syntactic Analysis

This stage involves parsing the various words that have been identified into larger structures that extract the relationships that exist between these. This might involve picking out some name as being the subjects of relationships, extracting any given details of the number or gender of this entity, and then locating the associated verb group, along with its tense and voice. Additional information may also be extracted, depending on the particular application. Typically, this will be done using an approximate 'shallow' parser, rather than making an attempt to accurately parse the whole text, using a restricted grammar and heuristics, perhaps searching for a restricted number of task-dependent verbs, and ignoring sentence fragments referring to other actions (as seen in several of the KA systems described in section 3 above).

Coreference Resolution

A subtask that lies between the syntactic and domain analyses is that of coreference resolution. This is the task of recognising the common reference shared by a number of entity names — for example, that the names "International Business Machines", "IBM" and "the computer company" in a particular article all refer to the same entity. This task also involves resolving instances of anaphora, temporal references and, perhaps, recognising whole-part, set-subset, etc. relationships between named entities.

The knowledge engineering approach to this task (which is that adopted by most) involves developing a heuristically driven module that will select some name (noting its type, number, gender, etc.), determine the scope of the name (that is, the extent of text within which references to this entity might be expected to occur), select candidate names within this scope and perform some syntactic and semantic checks (perhaps using domain ontologies) to decide if the same concept is being referred to. (This problem does not seem to lend itself naturally to a statistical approach. Several attempts have been made using supervised learning (e.g., the decision-tree building approach of McCarthy and Lehnert (1995)), but with limited success.)

At MUC-6, operating on a constrained problem, the performance of the better systems lay in the 51%–63% range for recall, and the 62%–72% range for precision (compared with values of 80% recall and 82% for humans attempting the same task) (Sundheim, 1996). At MUC-7, UPenn's system achieved a precision of .8 (but this was accompanied with a recall of only .3).

Domain Analysis

The final stage involves binding the individual scraps of information extracted during the previous stages into fuller descriptions of the particular event(s) to which the document refers. This can prove a difficult task for humans; figures of 60%–85% agreement between human subjects have been reported.

In general, these descriptions will be based upon some event template supplied (in explicit or implicit form) by the client. This template will specify the information that is requested, along with the format in which it should be presented — typically, this will be a database-record schema. It is necessary to balance several not necessarily compatible influences when devising the template: the template as a tool for representing the relevant sort of events; the

template as tool for easing the extraction of information from relevant documents, and; the intended use of completed templates (which might be used to generate reports to human operators, or as input to another computer system).

A knowledge engineering approach is generally adopted when building a component to address this task. This approach involves examining a set of texts in the domain of interest, identifying within these common, reliable patterns of words which convey the desired information (and which are expressible using the class of grammars to be used), writing rules to extract this information and place it in a template, and, finally, trying to devise some heuristics that will catch less common or less reliable patterns. As might be expected, the domain analysis stage is heavily domain-dependent and application specific, and systems cannot easily be converted to perform this analysis for different classes of events. Possible solutions to this difficulty might lie in developing (shallow) coverage of broad domains, in devising automatic learning strategies to support/replace the hand-writing of rules, or else in building general purpose text understanding systems.

The state of the art for this task is an F-measure value of about .56 (Appelt and Israel, 1999); however, as might be expected, there is some difficulty in deciding what information ought to be (and, indeed, can be) extracted during this task. For the purposes of evaluation at the MUCs, this phase is considered to consist of three separate sub-tasks:

- *Template element production* — relevant entities are extracted from the text and, along with other information acquired from the text, are used to instantiate templates. At MUC-7 the best system achieved an accuracy of 80% (compared to 93% achieved by humans performing the same task).
- *Template relation production* — involves identifying and extracting the relationships that exist between entities in the text. The best scores at MUC-7 were around 75%.
- *Scenario template production* — the elements and relations are combined within a template representing the events described in the text. The best scores for this phase are around 60% (Marsh, 1998; Appelt and Israel, 1999).

4.2.2 Information Extraction and Knowledge Acquisition

As should be evident from the above discussion, IE can be seen as a form of KA: a document (which may be, say, a transcription of an interview with an expert or a textbook) is processed to extract information that is structured using some template (which might well be a rule template, for example, rather than a database-record template). Indeed many of the ideas and techniques occurring in IE systems can be seen to have parallels in the sorts of approaches that have thus far been adopted to try to automate KA from NL. As with the KA tools, a major strength of IE systems is that, while they may not be as accurate as humans, they are usually a lot quicker. And, as might be expected, the difficulties for both tend to centre upon the semantic processing of their inputs. As a number of researchers seem to have realised, the worth of current systems, of both types, may lie in the assistance that they are able to provide for humans.

4.3 Information Retrieval

Information Retrieval (IR) involves selecting some relevant subset of a library of documents in response to a user's request. In contrast to the IE task, there is no attempt to extract and formalise the information contained within these documents — it is the documents themselves that are presented to the user.

The key notion here is that of *relevance* — the IR task is to retrieve as many relevant, and as few irrelevant, documents as possible (Baeza-Yates and Ribeiro-Neto, 1999). This field has existed for some time, but the growth of the WWW in the last decade has led to a renewed interest — and greater need for — IR.

Before retrieval can begin, it is necessary to define the scope of the search; that is, the document database. Since it is likely that this will be very large, the next task is to construct some sort of index over the database that will allow retrieval to be performed in a reasonable amount of time. A common approach to indexing is to build an inverted file of the vocabulary used, with, for each element of this vocabulary, the list of the documents in which it occurs. A retrieval episode is initiated by the user providing some sort of statement of information need. This might be in the form of, for instance, a natural language statement describing the domain of interest and the specific information required about this domain. This statement must then be translated into a query that the IR system can relate to its index and database. Currently, this translation of statement into query can be performed manually (as is the case with most current WWW search engines) or automatically, or in some semi-automatic fashion. The IR system can now process the query and attempt to retrieve relevant documents. Typically, the system will associate with each retrieved document some measure of the likelihood that the document meets the query; these allow the documents to be ranked according to their relevance. There may also be some sort of feedback loop in the process, whereby some (human or automatic) agent acts to modify the query based on the results that are being returned.

In general, most systems use some frequency-based measure to retrieve documents, with certain combinations of words suggesting relevance. The words provided in the query may be augmented by the use of thesauri, or ontologies and other background knowledge to provide a richer description of the target concepts, but the use of natural language processing techniques has not, in general, led to a significant improvement in performance (Harman, Schäuble and Smeaton, 1998).

The annual Text Retrieval Conferences (TRECs), sponsored by NIST and DARPA, provide a forum at which systems can be evaluated and compared (the eighth TREC took place in November 1999). As with the MUCs, the TRECs are based around a number of tests, which give some indication of the relative precision and recall capabilities of the competing systems.[5]

There are a number of different testing 'tracks', focusing on different aspects of information retrieval, and the testing mechanisms rely on a complicated pooling system (since, given the quantity of documents involved, it is unfeasible for a human to perform the tasks manually and provide 'ground truths' for the tasks — see (Voorhees and Harman, 1999) for details of the testing method). The results of the *ad hoc* track (that which corresponds most closely to the conventional view of IR) at TREC-8 indicate that the best systems are those with some element of human intervention (although the differences in the type and extent of human intervention among systems undermines, to a certain degree, the notion of truly comparative testing), with precision values of around 60%, although most systems produce results of less than 50% precision (human performance on this task, involving reasonable (but not negligible) amounts of effort are thought to be around 70%) (Voorhees and Harman, 1999; Sparck Jones, 1999). However, recent conferences have seen a levelling-out of increases in precision and a certain stagnation in approach (to such an extent that the decision

[5] *Precision*, in this context, being the proportion of retrieved documents that are relevant, and *recall* the proportion of relevant documents that are retrieved.

has been made to omit the *ad hoc* track from the next TREC). Since the results generated by the systems are far from perfect, it remains to be seen whether some sort of theoretical barrier has been reached, or whether some new methodology is needed to improve retrieval.

It should be noted that this discussion of IR has centred on text-based IR, since this is the area upon which most of the research work has been focused; it is expected that multimedia IR (which may include the retrieval of spoken information), will achieve increased prominence in the coming years.[6]

4.3.1 Information Retrieval and Knowledge Acquisition

There would seem to be some scope for applying IR within the KA process. An IR system (with, perhaps, some degree of human control in the search) provided with some notion of what might constitute a relevant knowledge-containing document in some domain might offer a useful means by which a large collection of documents (so large as to make a manual examination of it impractical) could be pruned into a useful subset. This subset of documents could then be analysed, either by a human, or, perhaps, an IE system, and the contained knowledge extracted.

It does not seem, however, that current IR systems would be able to perform such an operation on a heterogeneous collection of documents (such as that represented by the WWW) since current systems provide less than perfect results and require tailoring for particular content and domains. But given a more uniform document collection (an organisation's formal archives, say) even a less than perfect system may offer a more efficient route towards KA.

4.4 Machine Translation

Machine Translation (MT) systems attempt to produce, or help a human to produce, translations of given texts into a second language. Two theoretical approaches to the MT task have emerged:

- the *transfer* approach — texts are translated directly into the target language.
- the *interlingual* approach — texts are first translated into some artificial language (an 'interlingua') from which they can then be translated into the target language.

While the former approach might seem to be the more natural, the interlingual approach promises greater robustness and efficiency in that, there would be a need for two systems only for each language – one to translate it into the interlingua, and the other to translate the interlingua back into the language.[7] Unfortunately, no-one has yet been able to devise a sufficiently expressive interlingua, which provides the principal objection to the interlingual approach.

As with IE systems, the more traditional methods of constructing MT systems tended to rely on a knowledge engineering approach (a notable example of which is the METEO system (Chevalier, et al., 1978)), while, more recently, emphasis has shifted onto the construction of systems using the automatic exploitation of data resources. In the case of MT, such resources are, in general, in the form of existing translations — these may be used to learn statistical information about the task (e.g. (Brown et al., 1990)) or else in a case-based reasoning-like fashion make use

[6] Recent TRECs (from TREC-6 (1997) onwards) have included a 'spoken document retrieval' evaluation track.

[7] Hence, if there were *n* languages of interest, a total of *2n* systems would be needed to implement translation between each language. With the transfer approach, on the other hand, a total of *n(n-1)* systems would be needed to accomplish this.

of previous examples of translation (Sato, 1992). These alternative methods have served to raise the question of the extent to which knowledge other than that of linguistics is needed for this task; it would seem that some deeper understanding of the text and some concept of the context in which the text arises is necessary in order to select an appropriate translation. However, it might be that enough of the context can be inferred from the text alone to produce reasonable translations without the need to encode a vast store of background knowledge.

Unlike IE, there does not exist a well-established method of evaluating MT systems. The general LE criteria of adequacy, diagnostic and performance evaluation hold for MT systems, but within these there would seem to be a need for incorporating measures of the type and quality of the input, the accuracy and fluency of the output and the productiveness of the system when compared to human translators, and so on. The performance of current MT systems, regardless of implementational approach, is disappointing, with the best systems displaying an accuracy of perhaps about 80% on restricted problems (for example, the hybrid system developed as part of the Verbmobil project[8] produces 'approximately correct' translations from German to English (spontaneous speech, 2500 word vocabulary) with a reported accuracy of 74.2%), and, despite the emergence of several systems and services intended for public use,[9] there is no real likelihood of machines capable of full, reliable translation in the near future (Kay, 1998). One promising future direction is suggested by the relative success of IE systems — a practical approach to translation may involve extracting the essentials of the information within the text and then translating the formal representation of this information into the target language. Another approach might be to develop models of collaborative translation, whereby an initial, machine translation is post-edited by a human (whether such an approach would ever be more effective than direct human translation, however, remains an open question).

4.4.1 Machine Translation and Knowledge Acquisition

The usefulness of MT for KA applications would seem to lie in its potential for giving access to knowledge-bearing documents written in a foreign language. (Once translated, these documents could then be processed by a human knowledge engineer, or even an automated IE/KA system.) However, the current performance levels of current MT technology, and the high quality of translation that would (presumably) be needed if accurate knowledge were to be extracted would seem to suggest that such an approach would only be practical, if at all, if the documentation in the foreign language was so abundant and rich (and the knowledge sources in the native language so poor) as to make it the primary source of knowledge.

4.5 Speech Recognition

Speech Recognition (SR), as considered here, is the task of converting some input acoustic signal into the set of words which it represents. The ability to convert automatically spoken text to written would be of obvious practical benefit; accordingly, much research has been devoted to this, and a number of systems have been constructed. Zue, Cole and

---

[8] http://verbmobil.dfki.de/overview-us.html.

[9] For example, AltaVista's on-line translation service (http://babelfish.altavista.digital.com). On the whole, the translations provided by this, and other available systems, will be flawed, but from which, nonetheless, a user will often be able to infer the meaning; the usefulness of these tools for non-critical communication should perhaps not be underestimated.

Ward (1998) describe a set of parameters that can be used to describe these systems, and then compare their abilities. These parameters also give an insight into the current issues that surround SR tasks:

- *speaking mode* — this can range from identifying isolated words to processing continuous speech.
- *speaking style* — either scripted (i.e., the text is written down, and read to the system) or spontaneous (which will presumably contain (more) grammatical errors, pauses, etc.).
- *enrolment* — whether the system is tailored toward one particular speaker (and as such, may have been trained upon this speaker's speech patterns) or is speaker-independent.
- *vocabulary* — ranging from small (less than 20 words) to large (greater than 20000).
- *language model* — either finite-state (i.e., the language is restricted so that the permissible choices for the next word are known explicitly) or context sensitive, which more closely approximates to natural language.
- *perplexity* — the geometrical mean of the number of other words that can follow a particular word, given the language model and vocabulary, ranging from small (less than 10 words) to large (greater than 100).
- *signal-to-noise ratio* — as might be expected, SR system performance deteriorates under noisy conditions.
- *transducer* — which may be anything from a high-quality microphone to a standard telephone connection.

In addition to these factors, speech phenomena such as phonetic variance (differences in the pronunciation of a phoneme at different positions in speech), inter-speaker variance (differences of pronunciation from one speaker to the next) and intra-speaker variance (differences of one speaker's pronunciation, due to emotional state, etc) serve to make speech recognition in its most general form a very difficult task.

In common with many areas of LE, the early attempts to develop SR systems were primarily knowledge engineering approaches, relying on encoding expert knowledge of language. However, the successes of this approach were few, and it was not until the adoption of statistical approaches (over the last decade or so) that significant advances in performance have been made (Cunningham, 2000). In particular, Hidden Markov Models are widely used in speech recognition systems, trained using large numbers of annotated speech data.

The performance of SR systems is typically expressed in terms of *word error rate*, *E*;

$$E = \frac{S + I + D}{N} \times 100$$

where *N* is the total number of words in the test set, and *S*, *I* and *D* are the total numbers of substitutions, insertions and deletions respectively. Zue, Cole and Ward (1998) observe that *E* falls by a factor of 2 every two years, and go on to suggest that this is due to a combination of the advent of HMMs, the development of large corpora, the establishment of more rigorous testing procedures, and faster computer hardware. They offer the following examples to give some indication of the state of the art:

- *digit recognition* task — recognising strings of digits (continuous speech, speaker-independent, telephone bandwidth, perplexity = 11): $E = 0.04$ (when string length known in advance).
- *resource management* task — recognising enquiries about naval manoeuvres (speaker-independent, 1000 word vocabulary, perplexity = c.60): $E < 4$.
- *dictation* task — speaker-independent, 20000 word vocabulary, continuous speech, perplexity = c.200: E = around 7 (restricted domain).

These results sound impressive, and indeed there have been a number of commercial applications of speech recognition, in telephony and in home dictation packages. However, current continuous speech systems can achieve

speaker independence only at the price of domain dependence (Hausser, 1999). Zue, Cole and Ward strike a note of caution:

> "Even though much progress is being made, machines are a long way from recognizing conversational speech. [...] It will be many years before unlimited vocabulary, speaker-independent continuous dictation capability is realized."

4.5.1 Speech Recognition and Knowledge Acquisition

One obvious application of SR in KA would be as a tool for automatically transcribing (recordings of) expert protocols or interviews between domain experts and knowledge engineers; the transcription could then be analysed, either manually or perhaps by an IE system. Transcription can prove an extremely time-consuming task when done manually.

Again, the question is now one of whether the SR technology is mature enough for an application such as this. Most current useful applications of SR expect a single speaker (perhaps speaking continuously) using a restricted vocabulary and grammar. In contrast, during protocols/interviews, speakers are free to make even ungrammatical statements (which may yet be understood) and use a wide vocabulary (which may in practice be restricted somewhat by the domain, and the particular topic of the interview, but not necessarily so). Any attempt to constrain the content or manner of the participants' speech would mean that they were no longer using NL, and so, may serve to undermine the process by stifling the ability of the experts to express their knowledge. Furthermore, interviews consist of (usually) two speakers participating in a dialogue, which would seem only to compound the difficulties faced by SR systems.

Hence, it seems unlikely that a SR could cope with this task at the current time, especially when one considers that problems for humans arise when the recording is indistinct, or noisy, and some interpretation must be applied to complete the transcription. Other potential applications of SR in KA from NL would seem to be limited since existing records tend to be textual rather than spoken.

4.6 Natural Language Generation

A Natural Language Generation (NLG) system attempts to construct comprehensible natural language from its internal representation of information. Generally, this will involve a number of sequential sub-tasks (Hovy, 1998):

1. Some situation or event initiates the generation of NL; the goals of the utterance to be created, and the information on which it is to be based are identified.
2. High-level text planning is performed to devise a coherent structure for conveying this information.
3. Lower level sentence planning is performed to organise the structure of the individual sentences within the wider text.
4. Surface realisers formulate grammatically correct portions of text to build these sentences.
5. The generated text is delivered to the user, perhaps through the medium of a speech synthesis program.

There are a number of different strategies, which vary in sophistication and complexity, that can be adopted to address this task. The most basic approach is to use 'canned text', pre-built sentences that are simply printed for the user when some general event is encountered, but with no alterations made to account for the particular context in which the system and its user find themselves. The error messages generated by computer applications are typical examples of this sort of NL generation; the generality of the messages can render them inappropriate or unhelpful to the user.

The next level of sophistication involves basing the text around a number of template structures. This technique is useful when repeated utterances of the same general form need to be made, but with minor differences in content. (Among the interview-type systems for KA mentioned above are typical examples of this sort of approach to generating language. In their dialogues, for example, the same questions are asked about different domain concepts; in each case, the name of the concept is inserted into a standard template to form the new question.) The template approach has found some use in practical applications, ranging from compiling stock market reports from news wires, to preparing weather forecasts from weather maps.

Systems operating at the next level of sophistication use a phrase-based approach, in which grammar rule-like transformations (at the sentence level) and general text plans (at the document level) are used as types of generalised templates, in which the different parts of speech, such as *noun phrase*, *verb*, etc. are resolved into appropriate words. (This approach can be extended further with the inclusion of *features* within the grammar (e.g. the number of a noun or the tense of the verb) which have to correspond, and the incorporation of elementary semantics (such as whether a verb must be performed by an animate agent). On the whole, this approach still seems to be at the research stage, particularly at the document level, which lacks the foundation provided by well-defined language grammars at the sentence level; accordingly, this technique has mainly been applied to the task of generating isolated sentences (Hovy, 1998).

A closely related topic in LE research is *dialogue management*; the implementation of some system that is able to enter into and maintain a purposeful dialogue to some end would be of obvious benefit to those who have to work and interact with computers. There have been two main theoretical frameworks for viewing the nature of dialogue. The first, *discourse analysis*, views dialogue as an act of rational cooperation towards some common goal, and considers sentences to be well-formed. The later theory of *conversational analysis* recognises that spoken human dialogues are rarely this well-behaved, and tries to accommodate abrupt shifts of focus, ungrammatical utterances, and the like. Often, systems are developed with the aid of large corpora of annotated dialogue examples.

A dialogue manager may be thought of as consisting of two separate components (Giachin, 1998):

- An *interaction history* is used to record the progress of the dialogue; this is important, since the composite 'meaning' of the dialogue is distributed across the utterances which constitute it, and also because precise meaning of utterances may only be clear in the context of the previous remarks or later information.
- An *interaction manager* is used to control the dialogue (which might be driven by the user, or else, the onus is placed upon the system to develop the interaction).

Attempts to judge the worth of these systems are not always easy since they are rarely, if ever, found in isolation. A number of criteria can be considered, most of which are related to the adequacy of the system for some particular interaction, and are subjective to a certain degree, and may include a rating of the clarity and naturalness of the system's dialogue, the robustness of the system, and so on. The following set of core metrics were identified in the SUNDIAL project (Peckham, 1993; Simpson and Fraser, 1993):

- *Transaction success* - this metric measures how successful the system has been in providing the user with the requested information.
- *Number of turns* - this is a measure of the duration of the dialogue in terms of the number of turns taken to complete the transaction.

- *Correction rate* - this is a measure of the proportion of turns in a dialogue that are concerned with correcting either the system's or the user's utterances, which may have been the result of speech recognition errors, errors in language understanding, or misconceptions.
- *Contextual appropriateness* - this is a measure of the extent to which the system provides appropriate responses.

Current dialogue systems are very much domain- and task-dependent, answering questions about, for example, train timetables.

It is difficult to reach any conclusions about the state of the art in NL generation, since the systems can often only be judged according to their adequacy in a particular context (e.g. in providing informative error messages to the user of some application) and their performance relies to a great extent upon that of other technologies, such as speech recognition and natural language understanding. However, there do seem to be some significant gaps in current knowledge that would prevent a full and flexible language generation system from being constructed. These problems occur at the levels of lexical choice (e.g., whether "IBM", "the computer company" or "it" is most appropriate in a given context), of sentence planning (how a sentence is to be structured to best convey the desired meaning), of paragraph planning (how a number of sentences are to be structured and ordered to best convey the meaning), and also at the level of the underlying knowledge representation (what representation of meaning is appropriate to the construction of language expressing this meaning) (Hovy, 1998; van Noord and Neumann, 1998; Bateman, 1998). When one also considers the problems of style and rhetoric, the choice of an appropriate register for expression, it seems that human-like NL generation is still a distant prospect.

4.6.1 Natural Language Generation and Knowledge Acquisition

A major use for NL generation systems in knowledge acquisition would seem to lie in automating (aspects of) interviews with domain experts, an approach already seen, described in section 3.1. A system that was able to interrogate an expert for his/her knowledge, and then encode the replies appropriately in its own knowledge base would represent a more efficient method of KA, in which there would be no need for the time-consuming intermediary work of the knowledge engineer.

However, the approaches along these lines have generally used a restricted template-based question and answer approach. Such an approach relies on both the types of question that will be asked and the types of answer that will be provided being quite well-defined beforehand; this was achieved by placing the interview session in the context of partial domain knowledge, and assuming that any alterations to the knowledge base can be accommodated within the existing representation and problem-solving paradigms. Hence, such an approach would appear to be more useful during the later stages of knowledge base development and refinement, and for maintaining the system in its operational environment.

Whether the current technology could be extended beyond this is unclear, and depends to a great extent upon the maturity of other language technologies and on approaches to knowledge base representation and maintenance (the previous interview systems appear to have been built 'on top' of an existing knowledge base system structure, rather than being developed and implemented in parallel, which seems to limit the scope for changes that can be made). Nevertheless, even if constrained to focusing a subset of a knowledge base this approach remains a very appealing one, one which might prove a fruitful area for further investigation.

4.7 Natural Language Understanding

The task set a Natural Language Understanding (NLU) system is generally thought of in terms of translating some NL input into a description which encapsulates its meaning, and in such a way that the description can be tested against (the system's knowledge of) the real world in order to determine its degree of truth. Since NL is ambiguous, in general this process will require a reduction of the input into a rigid semantic language, the terms of which have well-defined truth conditions (Pulman, 1998 - usually, this language will be some sort of propositional logic). So, while this task is in some ways similar to the IE task (inasmuch as it involves re-describing the content of the input into some other, more formal representation), the aims of the two fields are rather different: the goal of NLU is to extract the underlying meaning of the text, rather than to extract the information contained within it.

To perform this reduction of the text into its logical meaning, some mapping from the words and syntax of natural language onto the expressions of the semantic language is needed. In the course of this, in general, a 'deeper' parse of the text will be needed than is considered to be necessary for IE. Even for the simplest of texts, extracting its literal meaning will rarely be a straightforward exercise, and tropes such as irony, metaphor and metonymy subvert and shift meaning, adding layers of complexity to the task of understanding. Allen (1995) highlights the different areas of knowledge that must be brought to bear upon this task:

- Phonetic/phonological knowledge - how sounds relate to words.
- Morphological knowledge - how words are constructed from their primitive elements.
- Syntactic knowledge - how proper sentences are formed.
- Semantic knowledge – what words mean, and how these meanings are combined to give the meanings of sentences and larger units of text.
- Pragmatic knowledge – how words are used in different contexts, and how their use and context affects their meaning.
- Discourse knowledge– how preceding sentences affect the interpretation of the next sentence (see (Scott and Kamp, 1998) for more about the modelling of discourse).
- World knowledge – including knowledge of own and others' beliefs and goals.

The range and complexity of this knowledge should indicate how difficult the task of NLU is. The problems encountered by IE systems discussed above are compounded in the case of NLU systems, since these attempt to resolve the input in a much more rigorous fashion. It is difficult to judge the performance of current technologies, but when one considers that there is no consensus on the meaning of understanding (and of meaning itself), it seems likely that a complete NLU system, operating with human-like competence remains a distant prospect.

4.7.1 Natural Language Understanding and Knowledge Acquisition

Obviously, the ability for NLU would mean that most, if not all, of KA from NL could be automated. Provided with some transcription of an interview or protocol, or, in tandem with other LE technologies, performing direct interrogation of the expert, a NLU system would have the means to encode directly the expert's knowledge, and in a form that would allow processing by an intelligent system. However, since this area of study is not well understood, and current NLU systems lack both coverage and scalability (Pulman, 1998), the implementation of such an approach for KA does not seem to be feasible in the immediate future.

5 Conclusions

This document represents an attempt to provide some overview of the current level of maturity of technologies that have been or, in the case of LE topics, have the potential to be applied to the task of acquiring knowledge from NL. It seems fair to say that the automation of KA from NL is still at the research stage: most techniques require some degree of human assistance (and a collaborative system, exploiting the complementary strengths of human and computer, perhaps represents the most feasible approach to this task in the immediate future), and there is little or no practical application of systems as yet. Consequently there is much scope for developing both the techniques themselves and the underlying 'theory' of KA from NL.

References

Appelt, D. E & Israel, D. J. (1999). Introduction to Information Extraction Technology. Tutorial prepared for the *International Joint Conference on Artificial Intelligence, 1999 (IJCAI-99)*. Available on-line at: http://www.ai.sri.com/~appelt/ie-tutorial/IJCAI99.pdf.

Aussenac-Gilles, N., Biébow, B. & Szulman, S. (2000). Revisiting ontology design: a methodology based on corpus analysis. In . In Rose Dieng & Olivier Corby (eds.), *Knowledge Acquisition, Modeling and Management, Proc. Twelfth European Knowledge Acquisition Workshop (EKAW-2000), Juan-les-Pins, France, October 2-6, 2000*, Lecture Notes in Computer Science, Vol. 1937, Springer, 2000 pp. 172-188.

Baeza-Yates, R. & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley Longman Pub., 1999.

Bateman, J. (1998). Deep generation. In Cole, R. A., et al. (eds.), *Survey of the State of the Art in Human Language Technology*, Cambridge University Press, Cambridge, UK, 1998. Available on-line at: http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html.

Belkin, N. J., Brooks, H. M. & Daniels, P. J. (1988). Knowledge elicitation using discourse analysis. In B. R. Gaines and J. H. Boose (eds.), *Knowledge-Based Systems, Vol. 1: Knowledge Acquisition for Knowledge-Based Systems*, Academic Press, London, 1988.

Berry, D. C. (1987). The problem of implicit knowledge, *Expert Systems*, 4(3), 144-151.

Biébow, B., & Szulman, S. (1993). Acquisition and validation: from text to semantic network. In N. Aussenac, et al. (eds.), *Proc. Seventh European Knowledge Acquisition Workshop, EKAW-93, Toulouse and Caylus, France, September 1993*, Springer-Verlag, Berlin. pp. 427-446.

Biébow, B., & Szulman, S. (1999). TERMINAE: a linguistic-based tool for the building of a domain ontology. In Dieter Fensel, Rudi Studer (eds.), *Knowledge Acquisition, Modeling and Management, Proc. 11th European Workshop, EKAW '99, Dagstuhl Castle, Germany, May 26-29, 1999*, Lecture Notes in Computer Science, Vol. 1621, Springer, Berlin, pp. 49-66.

Blythe J. & Ramachandran, S. (1999) Knowledge acquisition using an English-based method editor. In *Proc. 1999 Knowledge Acquisition Workshop*, KAW99.

Boose, J. H. & Bradshaw, J. M. (1988). Expertise transfer and complex problems: using AQUINAS as a KA workbench for KBS. In B. R. Gaines and J. H. Boose (eds.), *Knowledge Acquisition for Knowledge-Based Systems*, Academic Press, London.

Bowden, P. R., Halstead, P. & Rose, T. G. (1996). Extracting conceptual knowledge from text using explicit relation markers. In N. Shadbolt, K. O'Hara & G. Schreiber, (eds.), *Proc. Ninth European Knowledge Acquisition Workshop (EKAW-96), Nottingham, UK, May 14-17 1996*, Springer-Verlag, Berlin, 1996, pp. 147-162.

Brill, E. (1994). Some advances in rule-based part-of-speech tagging. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Seattle, Washington.

Brown, P., Cocke, J., Della Pietra, S., Della Pietra, V. J., Jelinek, F., Lafferty, J. D., Mercer, R. L., & Roossin, P. S. (1990). A statistical approach to machine translation. *Computational Linguistics*, 16(2), 79–85.

de Chalendar, G. & Grau, B. (2000). SVETLAN' or how to classify words using their context. In Rose Dieng & Olivier Corby (eds.), *Knowledge Acquisition, Modeling and Management, Proc. Twelfth European Knowledge Acquisition Workshop (EKAW-2000), Juan-les-Pins, France, October 2-6, 2000*, Lecture Notes in Computer Science, Vol. 1937, Springer, 2000 pp. 203-216.

Chevalier, M., Dansereau, J. et al. (1978). TAUM-METEO: Description du Système, Université de Montréal.

Cole, R. A., Mariani, J., Uszkoreit, H., Varile, G. B., Zaenen, A. & Zampolli A. (eds.) (1998). *Survey of the State of the Art in Human Language Technology*, Cambridge University Press, Cambridge, UK.

Compton, P., Horn, R., Quinlan, R. and Lazarus, L. (1989). Maintaining an expert system. In J. R. Quinlan (ed.), *Applications of Expert Systems*, London, Addison Wesley, pp. 366-385.

Cunningham, H. (2000). *Software architecture for language engineering*. PhD Thesis, University of Sheffield, UK.

Davis, R. & Lenat, D. (1982). *Knowledge-Based Systems in Artificial Intelligence: AM and Teiresias*, McGraw-Hill, New York.

Delannoy, J-F., Cao, F., Matwin, S. & Szpakowicz, S. (1993). Knowledge extraction from text: machine learning for text-to-rule translation. In *Proc. of the Workshop on Machine Learning Techniques and Text Analysis, European Conference on Machine Learning (ECML-93)*, Vienna, Austria.

Diederich, J., Ruhmann I., & May, M. (1988). KRITON: a knowledge acquisition tool for expert systems. In B. R. Gaines and J. H. Boose (eds.), *Knowledge-Based Systems: Vol. 2: Knowledge Acquisition Tools for Expert Systems*, Academic Press, London, 1988.

Ejerhed, E & Church, K. (1998). Written language corpora. In Cole, R. A., et al. (eds.), *Survey of the State of the Art in Human Language Technology*, Cambridge University Press, Cambridge, UK, 1998. Available on-line at: http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html.

Ericsson, K. A. & Simon, H. A. (1984). *Protocol Analysis: verbal reports as data*, MIT Press, Cambridge, Mass.

Eshelman, L., Ehret, D., McDermott, J. and Tan, M. (1988). MOLE: a tenacious knowledge acquisition tool. In B. R. Gaines and J. H. Boose (eds.), *Knowledge-Based Systems: Vol. 2: Knowledge Acquisition Tools for Expert Systems*, Academic Press, London, 1988.

Faure, D. & Nédellec, C. (1999). Knowledge acquisition of predicate argument structures from technical texts using machine learning: the system ASIUM. In Dieter Fensel, Rudi Studer (eds.), *Knowledge Acquisition, Modeling and Management, Proc. 11th European Workshop, EKAW '99, Dagstuhl Castle, Germany, May 26-29, 1999*, Lecture Notes in Computer Science, Vol. 1621, Springer, Berlin. pp. 329-334.

Francis, W. & Kucera, H. (1982). *Frequency Analysis of English Usage*. Houghton Mifflin, Boston.

Giachin, E. (1998). Spoken language dialogue. In Cole, R. A., et al. (eds.), *Survey of the State of the Art in Human Language Technology*, Cambridge University Press, Cambridge, UK, 1998. Available on-line at: http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html.

Gillies, D. (1996). *Artificial Intelligence and Scientific Method*, Oxford University Press, Oxford.

Godfrey, J. J. & Zampolli, A. (1998) Language resources: overview. In Cole, R. A., et al. (eds.), *Survey of the State of the Art in Human Language Technology*, Cambridge University Press, Cambridge, UK, 1998. Available on-line at: http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html.

Goel, A., Mahesh, K., Peterson, J. & Eiselt, K. (1996). Unification of language understanding, device comprehension and knowledge acquisition, In *Proc. 10th Knowledge Acquisition Workshop, Banff, Canada, November 9-14, 1996*. Available on-line at http://ksi.cpsc.ucalgary.ca/KAW/KAW96/goel/kaw-final.html.

Gomez, F., Hull, R. & Segami, C. (1994). Acquiring knowledge from encyclopedic texts. In *Proc. of the ACL's 4th Conference on Applied Natural Language Processing, ANLP94*, Stuttgart, Germany, 1994. pp. 84-90

Gomez, F. (1995). Acquiring knowledge about the habitats of animals from encyclopedic texts. In *Proc. of the Workshop for Knowledge Acquisition, KAW-95, Banff, Alberta, Canada, 1995*, vol. 1, pages 6.1-6.22.

Grishman, R. & Calzolari, N. (1998) Lexicons. In Cole, R. A., et al. (eds.), *Survey of the State of the Art in Human Language Technology*, Cambridge University Press, Cambridge, UK, 1998. Available on-line at: http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html.

Hahn, U., Klenner, M. & Schnattinger, K. (1996a). Automated knowledge acquisition meets metareasoning: incremental quality assessment of concept hypotheses during text understanding. In *Proc. 10th Knowledge Acquisition Workshop, Banff, Canada, November 9-14, 1996.* Available on-line at: http://ksi.cpsc.ucalgary.ca/KAW/KAW96/hahn/kaw96.html.

Hahn, U., Klenner, M. & Schnattinger, K. (1996b). Automated knowledge acquisition meets metareasoning: incremental quality assessment of concept hypotheses during text understanding. In N. Shadbolt, K. O'Hara & G. Schreiber, (eds.), *Proc. Ninth European Knowledge Acquisition Workshop (EKAW-96), Nottingham, UK, May 14-17 1996*, Springer-Verlag, Berlin, 1996, pp. 131-146.

Hahn, U. & Schnattinger, K. (1997). An empirical evaluation of a system for text knowledge acquisition. In *Proc. European Knowledge Acquisition Workshop (EKAW-97)*, pp. 129-144

Hahn, U. & Schnattinger, K. (1998). Towards text knowledge engineering. In *Proc. Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, MIT Press, Menlo Park, California, pp. 524-531.

Harman, D., Schäuble, P. & Smeaton, A. (1998). Document retrieval. In Cole, R. A., et al. (eds.), *Survey of the State of the Art in Human Language Technology*, Cambridge University Press, Cambridge, UK, 1998. Available on-line at: http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html.

Hart, A. (1988). Knowledge acquisition for expert systems. In Göranzon, B. and Josefson, I. (eds.), *Knowledge, Skill and Artificial Intelligence*, Springer-Verlag, London.

Hausser, R. (1999). *Foundations of computational linguistics*. Springer-Verlag, Berlin.

Hirschmann, & Thompson, H. S. (1998). Overview of evaluation in speech and natural language processing. In Cole, R. A., et al. (eds.), *Survey of the State of the Art in Human Language Technology*, Cambridge University Press, Cambridge, UK, 1998. Available on-line at: http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html.

Hovy, E. (1998). Language generation: overview. In Cole, R. A., et al. (eds.), *Survey of the State of the Art in Human Language Technology*, Cambridge University Press, Cambridge, UK, 1998. Available on-line at: http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html.

Hull, R. and Gomez, F (1998). Automatic acquisition of historical knowledge from encyclopedic texts. In *Proc. Knowledge Acquisition Workshop, Banff, Canada, 1998.* Available on-line at: http://ksi.cpsc.ucalgary.ca/KAW/KAW98/hull/.

Kay, M. (1998). Machine Translation: the disappointing past and present. In Cole, R. A., et al. (eds.), *Survey of the State of the Art in Human Language Technology*, Cambridge University Press, Cambridge, UK, 1998. Available on-line at: http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html.

Kingston, J. (1994). Linking knowledge acquisition with CommonKADS knowledge representation. *Report AIAI-TR-156*, AIAI, University of Edinburgh.

Krupka, G. R. (1996). SRA: description of the SRA system as used for MUC-6. In *Procs. Sixth Message Understanding Conference, Columbia, Maryland, November 6–8 1995*, Morgan Kaufmann Pub., San Mateo, Ca, pp. 221–235.

Kupiec, J. (1992). Robust part-of-speech tagging using a Hidden Markov Model. *Computer Speech and Language*, 6.

Lamel, L. & Cole, R. (1998) Spoken language corpora. In Cole, R. A., et al. (eds.), *Survey of the State of the Art in Human Language Technology*, Cambridge University Press, Cambridge, UK, 1998. Available on-line at: http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html.

Lapalut, S. (1996a). Text clustering to help knowledge acquisition from documents. In N. Shadbolt, K. O'Hara & G. Schreiber, (eds.), *Proc. Ninth European Knowledge Acquisition Workshop (EKAW-96), Nottingham, UK, May 14-17 1996*, Springer-Verlag, Berlin, 1996, pp. 115-130.

Lapalut, S. (1996b). How to handle multiple expertise from several experts: a general text clustering approach. In F. Maurer (Ed.), *Proc. 2nd Knowledge Engineering Forum (KEF'96), Karlsruhe, January 1996*. Available on-line at: http://www-sop.inria.fr/acacia/personnel/lapalut/publications/ps-files/kef96.ps.gz .

Lebowitz, M. (1983a). Researcher: An Overview. In Proc. National Conference on Artificial Intelligence, Washington D.C., August 22-26 1983, Morgan Kauffman, Los Altos, pp. 232-235.

Lebowitz, M. (1983b). Generalisation from natural language text , *Cognitive Science*, 7(1), 1-40.

Lebowitz, M. (1988). The use of memory in text processing, *Communications of the ACM*, 31(12), 1483-1502.

Leonard, R. G. (1984). A database for speaker-independent digit recognition. In *Proc of the 1984 International Conference on Acoustics, Speech, and Signal Processing*, ?, 1984, pp. 42.11–14. Institute of Electrical and Electronic Engineers.

Lu, R. and Cao, C. (1990). Towards knowledge acquisition from texts. In B. Wielinga, J. Boose, B. Gaines, G. Schreiber, & M. van Someren, (eds.), *Current trends in Knowledge Acquisition*, IOS, Amsterdam, 1990, pp. 289-301.

Maedche, A. & Staab, S. (2000a). Discovering conceptual relations from text. In W. Horn (ed.), *Proc. Fourteenth European Conference on Artificial Intelligence (ECAI 2000), August 20-25 2000, Berlin, Germany*, IOS Press, Amsterdam, pp. 321-325.

Maedche, A. & Staab, S. (2000b). Mining Ontologies from Text. In Rose Dieng & Olivier Corby (eds.), *Knowledge Acquisition, Modeling and Management, Proc. 12th European Knowledge Acquisition Workshop (EKAW 2000), Juan-les-Pins, France, October 2-6, 2000*, Lecture Notes in Computer Science, Vol. 1937, Springer, Berlin. pp. 189-202

Marcus, M., Santorini, B. & Marcinkiewicz, M. (1993). Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2).

Marcus, S. (1988). SALT: A KA tool for propose-and-revise systems. In S. Marcus (ed.), Automating Knowledge Acquisition for Expert Systems, Kluwer Academic Pub., Boston, 1988.

Marsh, E. (1998). Overview of the results of the MUC-7 evaluation. In, *Procs. Seventh Message Understanding Conference, Washington DC, April 1998*, Morgan Kaufmann Pub., San Mateo, Ca.

McCarthy J. & Lehnert, W. (1995). Using decision trees for coreference resolution. *Proc. of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 1050–1055. Paper available on-line at: http://ciir.cs.umass.edu/pubfiles/jmccarthy_ijcai95.pdf.

McGraw, K. L. & Harbison-Briggs, K. (1989). *Knowledge acquisition: principles and guidelines*, Prentice-Hall, New Jersey.

Merialdo, B. (1994). Tagging English text with a probabilistic model. *Computational Linguistics*, 20(2), 155-172.

Mikheev, A. & Finch, S. (1995). A workbench for acquisition of ontological knowledge from natural text. In Proceedings of the 7th conference of the European Chapter for Computational Linguistics (EACL'95). Dublin, Ireland, 1995. pp. 194-201.

Motta, E., Eisenstadt, M., Pitman, K., West, M. (1988). Support for knowledge acquisition in the Knowledge Engineer's Assistant (KEATS). *Expert Systems*, 5(1), 6-28.

Motta, E., Rajan, T., Domingue, J. and Eisenstadt, M. (1990). Methodological foundations of Keats, the knowledge engineer's assistant. In B. Wielinga, J. Boose, B. Gaines, G. Schreiber, & M. van Someren, (eds.), *Current trends in Knowledge Acquisition*, IOS, Amsterdam, 1990, pp. 289-301.

Moulin, B. & Rousseau, D. (1992). Automated KA from regulatory texts, *IEEE Expert*, 7(5), 27-35.

van Noord, G. & Neumann, G. (1998). Syntactic Generation. In Cole, R. A., et al. (eds.), *Survey of the State of the Art in Human Language Technology*, Cambridge University Press, Cambridge, UK, 1998. Available on-line at: http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html.

Paul D. & Baker, J. (1992). The design for the Wall Street Journal-based CSR corpus. In, *Proc. Fifth DARPA Speech and Natural Language Workshop, Feb. 1992*, Morgan Kaufmann.

Peckham, J. (1993). A new generation of spoken dialogue systems: Results and Lessons from the SUNDIAL project. In *Proc. EUROSPEECH 93*, Berlin, pp. 33-40.

Price, P., Fisher, W. M, Bernstein, J. & Pallett, D. S. (1988). The DARPA 1000-word resource management database for continuous speech recognition. In *Proc. 1988 International Conference on Acoustics, Speech, and Signal Processing, New York, April 1988*, Institute of Electrical and Electronic Engineers, pp. 651–654.

Pulman, S. (1998). Semantics. In Cole, R. A., et al. (eds.), *Survey of the State of the Art in Human Language Technology*, Cambridge University Press, Cambridge, UK, 1998. Available on-line at: http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html.

Reimer, U. (1990a). Automatic acquisition of terminological knowledge from texts. In L. C. Aiello (ed.), *Proc. Ninth European Conference on Artificial Intelligence (ECAI-90), Stockholm, August 6-10, 1990*, Pitman, London, pp. 547-549

Reimer, U. (1990b). Automatic knowledge aquisition from texts: Learning terminological knowledge via text understanding and inductive generalization. In *KAW'90 - Proc. of the Workshop on Knowledge Acquisition for Knowledge-Based Systems*, pp. 27.1-27.16.

Sato, S. (1992). CTM: An example-based translation aid system using the character-based best match retrieval method. In, *Proceedings of the 14th International Conference on Computational Linguistics, Nantes, France, August 1992*.

Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van de Velde, W. & Wielinga, B. (2000). *Knowledge engineering and management — the CommonKADS methodology*. MIT Press, Cambridge, Mass.

Scott, D. & Kamp, H. (1998). Discourse Modelling. In Cole, R. A., et al. (eds.), *Survey of the State of the Art in Human Language Technology*, Cambridge University Press, Cambridge, UK, 1998. Available on-line at: http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html.

Shaw, M. L. G. & Gaines, B. R. (1988). KITTEN: knowledge initiation and transfer tools for experts and novices. In B. R. Gaines and J. H. Boose (eds.), *Knowledge-Based Systems: Vol. 2: Knowledge Acquisition Tools for Expert Systems*, Academic Press, London, 1988.

Simpson, A. & Fraser, N. (1993). Black box and glass box evaluation of the SUNDIAL system. In *Proc. EUROSPEECH 93, Berlin*, pp. 33-40.

Sparck Jones, K. (1999). Summary performance comparisons TREC-2 through TREC-8. In *Proc. Eighth Text REtrieval Conference (TREC-8), Gaithersburg, Maryland, November 17-19, 1999*, NIST Special Publication pp. 500-246. Elements of this publication are available on-line at: http://trec.nist.gov/pubs.html.

Sundheim, B. M. (1996). Overview of results of the MUC-6 evaluation. In *Proc. Sixth Message Understanding Conference, Columbia, Maryland, November 6–8 1995*, Morgan Kaufmann Pub., San Mateo, Ca, pp. 13–32.

Swanson, D. R. & Smalheiser, N. R. (1997). An interactive system for finding complementary literatures: a stimulus to scientific discovery, *Artificial Intelligence*, 91, 183-203. Available on-line at: http://kiwi.uchicago.edu/webwork/AIabtext.html.

Szpakowicz, S. (1990). Semi-automatic acquisition of conceptual structure from technical texts, *International Journal of Man-Machine Studies*, 33(4), 385-397.

Velardi, P., Pazienza, M. T., Magrini, S. (1989). Acquisition of semantic patterns from a natural corpus of texts, *SIGART Newsletter, Special Issue on Knowledge Acquisition*, No. 108, April 1989, pp. 115-123.

Virkar, R. S. & Roach, J. W. (1988). Direct assimilation of expert-level knowledge by automatically parsing research paper abstracts, *International Journal of Expert Systems*, 1(4), 281-305.

Voorhees, E. M. & Harman, D. (1999). Overview of the eighth Text REtrieval Conference (TREC-8). In *Proc. Eighth Text REtrieval Conference (TREC-8), Gaithersburg, Maryland, November 17-19, 1999*, NIST Special Publication 500-246. Elements of this publication are available on-line at: http://trec.nist.gov/pubs.html.

Winston, P. H. (1993). *Artificial Intelligence*, 3rd edition, Addison-Wesley, Mass., USA.

Zue, V., Cole R. & Ward, W. (1998). Speech recognition. In Cole, R. A., et al. (eds.), *Survey of the State of the Art in Human Language Technology*, Cambridge University Press, Cambridge, UK, 1998. Available on-line at: http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html.

Appendix A. Language Engineering Resources

An LE resource, as it is considered here, is any language information, knowledge, or program which is generally available and which may assist in the development, improvement or evaluation of a LE application. As such, this will usually be in a machine-readable form, but is not necessarily so. Cunningham (2000) draws the distinction between *algorithmic* resources and *data* resources:

- algorithmic resources are those that actually perform some sub-task in a LE process, such as a part-of-speech tagger, or a parser;
- data resources, on the other hand, supply the information that is needed in order to perform a sub-task (e.g., a lexicon of names used during the named entity recognition phase), or which is used to provide the evidence for the knowledge incorporated within an algorithmic resource (e.g., a text corpora used to train a HMM for named entity recognition).

In addition, there are several LE development tools and environments available, which may themselves be used for the preparation and management of other resources.

There has been a substantial increase in the number of resources available in recent years; however, that is not to say that these are sufficient. Godfrey and Zampolli (1998) attribute the lack of adequate resources for the majority of languages to the following factors:

- The tendency, continuing until relatively recent times, to test linguistic theories and programs using small amounts of putatively critical data;
- The high cost of creating linguistic resources.

The high cost of their development, in particular, has encouraged collaborative efforts to produce resources, and has strengthened the desire that, once created, these resources should be reusable across a range of applications.

Cunningham goes on to make the observation that while there is a healthy amount of reuse of data resources, there is little reuse of algorithmic resources, since the amount of work involved in adapting them to a new application and integrating them with other system elements remains prohibitive. The lack of a common representation language for communication between resources means that prospective users have been faced with the unappealing prospect of producing systems that either use another's representation (which may be unsuited to the application in hand), or else are able to translate between (perhaps several) different representations, with all the problems of maintaining coherence that this would entail. A number of suggestions have been put forward for what might be an appropriate common representation language; of these, SGML has probably achieved the widest acceptance (in the more recent MUCs, for example, SGML has been adopted as the language for specifying the inputs and outputs of IE systems and, elsewhere, the Expert Advisory Group on Language Engineering Standards (EAGLES),[10] an attempt to define standards for corpora, lexicons, grammars, and evaluation, settled upon formats based on SGML). In addition, the TEI (Text Encoding Initiative) has led to the development of a model for representing machine-readable dictionaries. In application systems, TFS (Typed Feature Structure) based formalisms are nowadays used in a large number of European lexical projects.

[10] http://www.ilc.pi.cnr.it/EAGLES/home.html.

A further, and perhaps more serious problem, which can afflict both algorithmic and data resources, surrounds the degree to which particular linguistic theories might be represented implicitly in the resource; this could effectively undermine the approach adopted by the re-user of the resource.

Despite these problems, a wide range of resources has become available, many distributed via the WWW. This appendix provides an (incomplete) overview of the types of resource that are available, listing a number of resources commonly referred to in the domain literature. More detailed and wider-ranging lists are maintained on-line by SIL.[11] Addresses for many of the resources mentioned here, and for many others, can be found in section 12.6 of (Cole et al., 1998).

A.1 Data Resources

A.1.1 Language Corpora

The importance of corpora has increased in recent years with the increasing tendency towards the use of statistical methods in LE. A corpus will consist of some collection of (written or spoken) documents and (usually) some sort of annotation. This annotation might be of, say, the parts of speech represented by the words in the corpus, or more complex parses of sentences. These data can then be used to develop and test language theories and models,

A.1.2 Written Corpora

An early million-word corpus, annotated with part-of-speech tags, and which instigated much research, was the Brown corpus (Francis and Kucera, 1982). Another widely used corpus, this time annotated with (amongst other syntactic and semantic information) parsing information is the Penn Treebank (Marcus et al., 1993).[12] The Linguistic Data Consortium, also at the University of Pennsylvania, distributes this along with a range of other corpora.[13] The International Corpus of English (ICE) is a long-term project to collect, analyse and then distribute texts in the various varieties of English spoken throughout the world.[14] This represents merely a small selection of the available written corpora resources; (Ejerhed and Church, 1998) provides an overview. In addition, there are a number of sources of electronic texts; a list of these may be found here.[15]

A.1.3 Spoken Corpora

If anything there would seem to be a greater need for spoken corpora, considering the scope for variation in pronunciation and quality. As for written corpora, the development of spoken corpora is on-going; in particular, there are a number of projects devoted to collecting examples of the various European languages.

Notable examples include the TI-DIGITS corpus (Leonard, 1984), a collection for digit recognition, the Resource Management corpus (Price et al., 1988) and the *Wall Street Journal* corpus (Paul and Baker, 1992) for continuous

---

[11] http://www.sil.org/linguistics/computing.html.

[12] See http://www.cis.upenn.edu/~treebank/home.html for more details.

[13] http://www.ldc.upenn.edu/.

[14] http://www.ucl.ac.uk/english-usage/ice/index.htm.

[15] http://www.sil.org/linguistics/ETEXT.HTML.

speech recognition. Many more collections, along with a more detailed discussion of this area, are described in (Lamel and Cole, 1998).

A.1.4 Lexicons

Here, a lexicon is some list containing information about words. This might be as simple as a common dictionary or thesaurus. Machine-readable dictionaries that have been widely used include the Longman Dictionary of Contemporary English, the Merriham-Webster Dictionary and the Oxford Advanced Learner's Dictionary.[16] However, these will usually be represented in terms of the morphological parts of words, which makes for more efficient processing, and may, in addition, include more detailed syntactic and semantic information. The WordNet project at Princeton University[17] is devoted to creating a record of word senses, and which also includes descriptions of part-whole, *is-a* and other relationships between words, so creating a partial ontological resource.

(Grishman and Calzolari, 1998) gives a more thorough overview of lexicon resources and their uses.

Lists of (company/proper/place) names, some task- or domain-specific, others more general, are frequently used, particularly by IE systems. Many such lists are available; the Computing Research Laboratory at New Mexico State University,[18] for example, provides a useful repository. Other sources, not dedicated to this field, offer potentially useful data: for example, the US Security and Exchange Commission's EDGAR database of trading company names, and governmental census data (for personal names).

A.1.5 Ontologies

Application independent ontologies such as WordNet and that of CYC Corporation[19] can be used to try to incorporate some semantic knowledge into systems. The use of ontological knowledge in LE is expected to increase as these resources become available.

A.2 Algorithmic Resources

As mentioned above, there are fewer algorithmic resources available, and those that are available tend to perform more basic LE tasks. This is due to the fact that the more complex the task, the more it tends to be domain- and task-specific, thus limiting the potential for its reuse. Listed below are some of the resources that are available; again, this list is not comprehensive, but is intended to give some indication of the sort of algorithmic resources that are available.

A.2.1 Taggers

Eric Brill's taggers are available on-line.[20] Xerox offers a tagging program.[21] The Corpus Research Group at the University of Birmingham offer a downloadable tagger[22]), and also an email tagging service.[23]

---

[16] A number of different on-line dictionaries/thesauri (including bilingual dictionaries, indispensable for machine translation) may be found at http://www.yourdictionary.com/ , along with other language resources.

[17] http://www.cogsci.princeton.edu/~wn.

[18] http://clr.nmsu.edu/.

[19] http://www.cyc.com.

[20] http://www.cs.jhu.edu/~brill/.

A.2.2 Parsers

A number of parsers are available; for example, the PROTEUS research team at NYU, has developed and made available a probabilistic chart parser, returning parses in the notation of the Penn Treebank,[24] and a team at Carnegie Mellon University has made available the parser it has developed, which promises to be robust and reliable enough for IE work.[25]

A.2.3 Morphological analysis

The Alvey Natural Language Tools set[26], developed by the Universities of Cambridge, Edinburgh and Lancaster, includes a morphological analysis component, as well as parsers, a grammar and a lexicon.

A.3 Development Environments

There are a several LE tool development environments available. A general aim shared by the developers of these environments is the encouragement of resource reuse, by reducing the overheads associated with the tedious job of making resources 'compatible'. These include:

- The General Architecture for Text Engineering (GATE), developed at Sheffield University,[27] consists of a graphical interface (allowing both data and system control visualisation), a document manager (easing the creation and manipulation of data resources), and a built-in collection of reusable algorithmic components for LE.
- The Multilevel Tools, Annotation Engineering (Mate) workbench,[28] developed through the collaborative efforts of a number of European research institutes, is specifically designed to aid in the display, editing and querying of annotated (using XML) speech corpora.
- Already mentioned above, the components of the Alvey Natural Language Tools set can be used in concert, integrated by a 'grammar development environment', to form a complete analytical system.

---

[21] ftp://parcftp.xerox.com/pub/tagger/.

[22] http://www-clg.bham.ac.uk/tagger/.

[23] Short, English emails sent to mtagger@clg.bham.ac.uk will be tagged automatically and returned.

[24] http://cs.nyu.edu/cs/projects/proteus/app/.

[25] http://bobo.link.cs.cmu.edu/index.html/.

[26] http://www.cl.cam.ac.uk/Research/NL/anlt.html.

[27] http://www.dcs.shef.ac.uk/research/groups/nlp/gate/.

[28] http://mate.mip.ou.dk/.