Knowledge-Based Systems
for Natural Language Processing

Kavi Mahesh
Sergei Nirenburg

MCCS-96-296

Computing Research Laboratory
Box 30001, Dept. 3CRL
New Mexico State University
Las Cruces, NM 88003-0001

Contents

Knowledge-Based Systems for Natural Language Processing

Kavi Mahesh and Sergei Nirenburg

Computing Research Laboratory
New Mexico State University
Las Cruces, NM 88003-8001
Ph: (505) 646-5466 Fax: (505) 646-6218
mahesh@crl.nmsu.edu sergei@crl.nmsu.edu

Abstract

This article reviews some of the underlying principles and methodological issues in developing knowledge-based methods for natural language processing. Some of the best practices in knowledge-based NLP will be illustrated through several NLP systems that use semantic and world knowledge to resolve ambiguities and extract meanings of sentences. Issues in knowledge acquisition and representation for NLP will also be addressed. The article includes pointers to books, journals, conferences, and electronic archives for further information. A revised version of this article will appear as a chapter in the *Handbook for Computer Science and Engineering* to be published by CRC Press in Fall 1996.

1 Introduction

A large variety of information processing applications deal with natural language texts. Many such applications require extracting and processing the meanings of the texts, in addition to processing their surface forms. For example, applications such as intelligent information access, automatic document classification, and **machine translation** benefit greatly by having access to the underlying meaning of a text. In order to extract and manipulate text meanings, a natural language processing (NLP) system must have available to it a significant amount of **knowledge** about the world and the domain of discourse. The knowledge-based approach to NLP concerns itself with methods for acquiring and representing such knowledge and for applying the knowledge to solve well-known problems in NLP such as **ambiguity** resolution. In this chapter, we look at knowledge-based solutions to NLP problems, a range of applications for such solutions, and several exemplary systems built in the knowledge-based paradigm.

A piece of natural language text can be viewed as a set of cues to the meaning conveyed by the text, where the cues are structured according to the rules and conventions of that natural language and the style of its authors. Such cues include the words in a language, their inflections, the order in which they appear in a text, punctuation, and so on. It is well known that such cues do not normally specify a direct mapping to a unique meaning. Instead, they suggest many possible meanings for a given text with various ambiguities and gaps. In order to extract the most appropriate meaning of an input text, the NLP system must put to use several types of knowledge including knowledge of the natural language, of the domain of discourse, and of the world in general.

Every NLP system, even a connectionist or ''purely'' statistical one, uses at least some knowledge

of the natural language in question. Knowledge of the rules according to which the meaning cues in the text are structured (such as grammatical and **semantic** knowledge of the language) is used by practically every NLP system. The term Knowledge-Based NLP System (KB-NLP) is applied in particular to those systems that, in addition to using linguistic knowledge, also rely on explicitly formulated domain or **world knowledge** to solve typical problems in NLP such as ambiguity resolution and inferencing. In this chapter, we concentrate on KB-NLP systems.

1.1 Knowledge-Based Ambiguity Resolution: Some Examples

Not all problems in NLP require world knowledge. For example, a sentence such as (1) below has several local syntactic ambiguities all of which can be resolved using grammatical knowledge alone. Sentence (2) has a lexical or **word sense** ambiguity where the word ''bank'' can mean either a financial institution or a river bank among other meanings. This ambiguity can be resolved using statistical knowledge that the financial institution sense is more frequent in the given corpus or domain. It can also be resolved (with a better confidence in the result, perhaps) using world knowledge that salaries are much more likely to be deposited in financial institutions in the modern world than in a river bank. Such knowledge is called a **selectional constraint** or preference.

Sentence (3) below also has a word sense ambiguity in the word ''bar'' which can mean (in its noun form) either a rod or a drinking place. The ambiguity can be resolved in favor of the rod meaning (such as in a ''chocolate bar'') in (3a) using world knowledge that a child is not allowed to inspect drinking places (or by knowing that the rod sense occurs much more frequently with ''child'' than does the drinking-place sense). Sentence (3b), however, is truly ambiguous and is hard to interpret one way or the other in the absence of strong knowledge of the domain of discourse.

Sentence (4) is an example of a semantic ambiguity in the preposition ''by.'' Once again, world knowledge about hiding events inform the system that the hat must have been hidden before the point in time known as ''three o'clock'' rather than by ''Mr. three o'clock'' or near the place known as ''three o'clock.''

> **(1)** The large can can hold the water.
> **(2)** The man deposited his salary in the bank.
> **(3a)** The child inspected the new bar.
> **(3b)** The officer inspected the new bar.
> **(4)** The hat was hidden by three o'clock.

1.2 Knowledge-Based Systems in NLP

A majority of KB-NLP systems are domain specific. They use rich knowledge about a particular domain to process texts from that domain. NLP is easier in domain-specific systems for several reasons: some of the ambiguities present in the language in general are eliminated in the chosen domain by overriding preferences for a domain-specific (or ''technical'') interpretation; the range of inferences that can be made to bridge a gap is narrowed by knowledge of the chosen domain; and it is much more tractable to

encode detailed knowledge of a narrow domain than it is to represent all the knowledge in the world. Domain-specific NLP systems can also benefit from expert knowledge about the domain which may be available in the form of rules, heuristics, or just episodic or case knowledge.

Some applications, however, do not allow us to constrain the domain narrowly. Often, any piece of text must be acceptable as input to the system or texts from a domain may nevertheless talk about pretty much anything in the world. In such tasks, since it is impossible to provide the system with expert knowledge in every domain, one must build a base of common-sense knowledge about the world and use that knowledge to solve problems such as ambiguity.

Knowledge-based solutions have influenced the field of NLP in significant ways providing a viable alternative to approaches based only on grammars and other linguistic information. In particular, knowledge-based systems have made it possible to integrate NLP systems with other knowledge-based AI systems such as those for problem solving, engineering design, and reasoning (e.g., Peterson et al., 1994). Efforts at such integration have moved the focus away from natural language front-ends to more tightly coupled systems where NLP and other tasks interact with and aid each other significantly. KB-NLP systems have provided a means for these AI systems to be grounded in real-world input and output in the form of natural languages.

KB-NLP systems have also brought to focus a range of applications that involve NLP of more than one natural language. Knowledge-based systems are particularly desirable in tasks involving multi-lingual processing such as machine translation, multilingual database query, information access, or multilingual summarization. Since linguistic knowledge tends to differ in significant ways from one language to another, such systems, especially those that deal with more than two languages, benefit greatly by having a common, interlingual **representation** of the meanings of texts. Deriving and manipulating such language-independent meaning representations go hand in hand with the knowledge-based approach to NLP.

Knowledge-based solutions are equally applicable to both language understanding and generation. However, knowledge-based techniques have been explored and applied in practice to language under-standing to a far greater extent than to generation. Although we use understanding as the task in this chapter to illustrate the majority of problems and techniques, KB-NLP has been quite influential in the field of generation as well.

2   Underlying Principles

NLP can be construed as a search task where the different possible meanings of words (or phrases) and the different ways in which those meanings can be composed with each other define the combinatorics of the space. All knowledge-based systems can be viewed as search systems that use different types of knowledge to constrain the search space and make the search for an optimal or acceptable solution more efficient. Knowledge such as that contained in selectional constraints aids NLP in two ways: it constrains the search by pruning certain branches of the search space that are ruled out as per the knowledge and it also guides the system in the remaining search space along paths that are more likely to yield preferred interpretations. Thus, knowledge helps to both reduce the size of the search problem

and to make the search in the reduced space more efficient.

In designing a KB-NLP system one must first determine how to represent the meanings of texts. Once such a meaning representation is designed, knowledge about the world, meanings of words in a language, and meanings of texts can all be expressed in well-formed representations built upon the common meaning representation (Mahesh and Nirenburg, 1996).

Knowledge in KB-NLP systems is typically partitioned into a **lexical knowledge** base and a world KB. A lexical KB typically contains linguistic knowledge, such as word meanings, the syntactic patterns in which they occur, and special usage and idiosyncratic information, organized around the words in the language. It is, in general, a good practice to keep language specific knowledge in the lexical KB and domain or world knowledge in a separate world KB which is sometimes also called an **ontology,** especially in multilingual systems (Carlson and Nirenburg, 1990; Mahesh, 1996). In such a design, the world KB can be common across all the natural languages being processed and can also be shared potentially with other knowledge-based AI systems. In what follows, whenever we refer to a KB or to retrieving knowledge from a KB, we mean the world KB, not a lexicon.

In some systems, such as case-based NLP systems (Cullingford, 1978; DeJong, 1979; 1982; Lebowitz, 1983; Lehnert et al., 1983; Ram, 1989; Riesbeck and Martin, 1986a; 1986b), in addition to ontological world knowledge, one or more episodic knowledge bases may be employed. An episodic KB, sometimes also known as an onomasticon or case library, contains remembered instances of ontological **concepts** (such as people, places, organizations, events, etc.), is typically used for case-based inferences but may also be used for selection tasks (see below).

The application of knowledge in KB-NLP systems can be divided into one of two fundamental operations:

- **Knowledge-based selection:** A very common problem in NLP is ambiguity where the system must choose between two or more possible solutions (meanings of a word, for instance). Knowledge-based selection is a method for choosing one (or a subset) of the possible meanings as per constraints derived from the system's knowledge of how meanings combine with those of other words in (syntactically related parts of) the text. Use of such selectional constraints for word sense **disambiguation** was illustrated in example sentences (2) and (3) above. Knowledge of the relationships between each of the possible meanings of a unit and meanings of other units of the text is used to set up a set of constraints for each possible meaning. These constraints are evaluated using the knowledge base and the results compared with each other. The choice that best meets all the constraints is selected by the system as the most preferred meaning of the unit.

- **Knowledge-based inference:** Often, there are zero choices to select from or there are gaps in the input where certain parts of the meaning or how certain parts compose with each other is not specified by any cues in the input. Knowledge-based systems apply their knowledge to infer such missing information to fill the gaps in the meaning. Knowledge-based inference is also often necessary to resolve a word sense ambiguity by making inferences about the meanings. For example, in the sentence ''The box was in the pen,'' an understander has to infer that the ''pen'' must have been a playpen (not a writing pen) using its knowledge of the default, relative sizes of boxes, writing pens, and playpens (unless it was known in the prior context that it was a playpen).

Many tasks, such as question answering and information retrieval, also require NLP systems to often make inferences beyond the literal meanings of input texts. The knowledge in the system permits such inferences to be made by means of reasoning and inferencing methods commonly employed in expert systems and other AI systems. Knowledge-based inferences are made essentially by matching partial meanings of input texts with stored knowledge structures and adding the resulting instantiations of the knowledge to the current interpretation of the text. Making such inferences is typically an expensive process due to the cost of retrieving the right knowledge structure from a KB and the typically large search spaces for making inferences from the knowledge structures. KB-NLP systems must often find ways to control inference processes and keep them goal-oriented.

Important issues in Knowledge-Based Selection include:

- It is often the case that the best alternative according to KB-selection is inappropriate. When such an error is revealed by later processing, the KB-NLP system must be able to <u>backtrack</u> and continue the search. It is also desirable to <u>recover from an error</u> by repairing the incorrect meaning rather than continue the search and reinterpret the text from the start. Various methods of error recovery have been implemented in different KB-NLP systems (e.g., Eiselt, 1989; Mahesh, 1995). Backtracking and failure recovery require the system to retain unselected alternatives for possible use later during recovery.

- The algorithm may in fact be made to eliminate those choices whose scores fall below a certain threshold to prune the search space and reduce the size of the search problem.

- KB-Selection might also use heuristics to guide the search just like in other AI systems. Examples of such heuristics include structural syntactic preferences (such as <u>minimal attachment</u> and <u>right association,</u> see below), those derived from knowledge of the domain, or other search heuristics.

- In addition to doing a best-first search and backtracking, the algorithm can also implement different schemes for dependency analysis to assign blame for an error and make an informed choice during backtracking. Such methods are warranted by the typically large search spaces encountered in trying to analyze the meanings of real-world texts with complex sentences containing multiple, interacting ambiguities.

- NLP involves the application of several different types of knowledge, including syntactic, semantic, and domain or world knowledge. Constraints from each knowledge source are satisfied to various degrees, resulting in different scores for each constraint. All of this evidence must be combined to make an integral decision in resolving an ambiguity. Since there is no optimal ordering for processing different types of knowledge that is always the right one for NLP, flexible architectures such as **blackboard systems** are ideally suited for building KB-NLP systems (Mahesh, 1995; Nirenburg and Frederking, 1994; Nirenburg et al., 1994; Reddy, Erman, and Neely, 1973).

- It is also possible for a selection algorithm to pursue a limited number of parallel interpretations and delay decisions until further information eliminates some of the meanings under consideration.

This approach leads to various problems in representing parallel interpretations and controlling their processing.

2.1   Methodological Issues

A basic requirement for KB-NLP is that the system must be provided with all the necessary knowledge. In spite of some advances in automated acquisition of knowledge for the purposes of NLP (Wilks, Slator, and Guthrie, 1995), acquisition must be done manually for the most part. As a result, KB-NLP systems share the knowledge engineering bottleneck with other knowledge-based AI systems. However, for many applications, the types of knowledge necessary for NLP make it easier to acquire the knowledge and to automate acquisition partially than in the case of problem solving or other expert systems.

There has been a lot of attention focused on the problem of designing a standardized knowledge representation language that serves as a common interchange format so that different systems and research groups can share knowledge bases developed by each other. Work on defining such standards is still in preliminary stages. However, several examples of sharable knowledge bases with translators between different formats can be found in experimental development already (Genesereth and Fikes, 1992; Gruber, 1993; IJCAI-Workshop, 1995).

Knowledge acquisition for KB-NLP involves building a broad coverage world model (or ontology, Carlson and Nirenburg, 1990; Knight and Luk, 1994; Mahesh, 1996; Mahesh and Nirenburg, 1995a) and a sufficiently large lexicon for each of the desired languages that represents meanings of words using the concepts in the world model. Any such effort to build a nontrivial KB-NLP system requires the collaborative efforts of a number of linguists, native speakers, lexicographers, ontologists, domain experts, and system developers and testers. In order to aid the acquisition of both the world model and the lexicons, it is essential to have a set of tools that support a number of activities including:

- searching and browsing the hierarchies and the conceptual meanings in the ontology;

- searching and browsing previously acquired lexical entries;

- searching and browsing an on-line dictionary, thesaurus, or corpus;

- graphical editing of the hierarchies and conceptual relationships in the ontology;

- lexical entry creation and editing (with minimal manual typing);

- various types of consistency checking both within the ontology and with the lexicon, and for conformance with overall design and guidelines;

- automatic testing and correction of ontological and lexical entries;

- supporting interactions between ontologists, domain experts, knowledge engineers, and lexicon acquirers, such as through an interface for submitting requests for changes or additions and for logging them; and

- database maintenance, version control, any necessary format conversions, and support for distributed maintenance and access.

In practice, a system never has all the knowledge that it might ever need. As such, it is important for KB-NLP systems to be able to reason under uncertainty and make inferences from incomplete knowledge. Since natural languages tend to have a large variety of constructs with many idiosyncrasies, it is invaluable for KB-NLP systems to be robust and be able to produce meaningful output even when complete knowledge is not available.

A drawback of the KB-NLP paradigm is that it is often intractable to provide the system with all the knowledge it may need. As a result, KB-NLP systems often fail to solve a particular instance of an NLP problem as they may not have the complete knowledge necessary for proper solution. There are two trends in current research aimed at solving this problem: multi-engine and human assisted systems.

Multi-engine systems use a KB-NLP engine as one of several engines that together solve the problem of NLP. For example, a KB-NLP system may work together with a statistical or a purely syntax-based system. In such systems, several designs are possible: a dispatcher could be used to examine each input and decide which engine to call; all engines could be run in parallel and an output selector used to select one or more of the outputs; or all engines could be run in parallel, in a race, and the first result used (Nirenburg and Frederking, 1994; McRoy and Hirst, 1990). In general, a dispatcher or output-selector architecture is easier to implement than one requiring the outputs of several engines to be combined.

Multi-engine architectures have been applied especially to machine translation among multiple natural languages (Nirenburg et al., 1992b; 1994; Frederking et al., 1993; Nirenburg and Frederking, 1994; Frederking and Nirenburg, 1994). Heterogeneous multi-engine architectures where different subsets of techniques and engines can be combined for different languages are particularly suited to multilingual NLP systems. They enable the overall system to have different sets of tools and engines for different languages. Such flexibility is essential for rapid development, since certain tools or engines may be too expensive to build for some languages but readily available for others.

Human assisted KB-NLP systems produce partial solutions to NLP problems and interact with a human user to incrementally perform the task. Depending on whether the human user is a naive user or an expert linguist, different modes and depths of interaction may be desirable. However, even when the users are expert linguists, these systems are very effective since they provide the experts with fast, sophisticated access to on-line dictionaries, lexicons, corpora, and so on and perform most of the less tricky computation very quickly. These systems also aid the users in visualizing problems, knowledge representations, and intermediate results with ease. With advances in hardware and software technology for building highly usable graphical interfaces, this alternative is becoming highly popular in situations that do not require fully automatic NLP. This also makes it possible to build integrated NLP workstations that support program development, debugging, testing, knowledge acquisition, and end-user interaction all in a single environment with immediate interactions and feedback between the different components (e.g., Nirenburg et al., 1992b; Nirenburg, ed., 1994).

3 Knowledge-Based NLP in Practice

Knowledge-based solutions have been applied to solve a variety of problems in NLP. In this Section, we illustrate some of the well-known problems and describe methods using knowledge-based selection to solve the problem. We also introduce the reader to several exemplary systems that have addressed the problems. These system descriptions also serve as a broad coverage of some of the most successful KB-NLP systems built so far. Systems using knowledge-based inference can be viewed as applications of NLP and will be described at the end of this Section.

3.1 Syntactic Ambiguity

A syntactic ambiguity can be of two kinds: category ambiguity and **attachment** ambiguity. Category (or part of speech) ambiguities often can be resolved using syntactic knowledge of the language. Attachment ambiguities often require semantic and world knowledge to resolve. A commonly occurring type of attachment ambiguity is the PP-attachment problem where there is an ambiguity in which part of a sentence a prepositional phrase (PP) is modifying. Constraints on composing the meanings of the two units being attached (see Figure 1) derived from semantic or world knowledge must often be applied to select one of the possible attachments.

A knowledge-based algorithm for resolving attachment ambiguities sets up constraints derived from semantic and world knowledge for composing the meanings of the child unit (being attached) with meanings of each of the possible parent syntactic units (at which the child could be attached). Such selectional constraints (e.g., Allen, 1987; Wilks, 1975) are typically represented in the form of a permissible range of fillers for slots in frames representing the meanings. The potential filler (i.e., the meaning of the child unit) is compared against this constraint by a fuzzy match function. A popular way to do this fuzzy match is to compute a weighted distance between the two meanings in a semantic or ontological network of concepts. The closer the two are in the network, the higher the score assigned to the particular choice. The algorithm then combines the scores from different constraints for the same choice by applying a mathematical function which may be as simple as addition or multiplication or may be complex variants of root-mean-squared functions and the like. The resulting combined scores are used to select the best choice according to the knowledge of selectional constraints (e.g., Beale et al., 1996).

For example, in Figure 1, the PP ''with the horse'' can be attached in any one of the three ways shown in the tree. Due to this ambiguity, it is not clear whether the telescope is accompanying the man or is an instrument of seeing. However, as shown in the lower half of the figure, knowledge of selectional constraints tells us that an instrument of seeing must be an optical instrument. Since a horse is not (typically) an optical instrument, the attachment to the verb phrase (VP) ''saw'' can be ruled out, resulting in an interpretation where the horse is an accompanier to ''the man.''

There have been a number of attempts, both in NLP and psycholinguistics, to develop purely syntactic methods for resolving syntactic attachment ambiguities. Several criteria, such as <u>minimal attachment</u> and <u>right association</u>, based on purely structural considerations have been proposed. While they work in many cases of attachment ambiguities, often they either lead to an incorrect choice (i.e., semantically
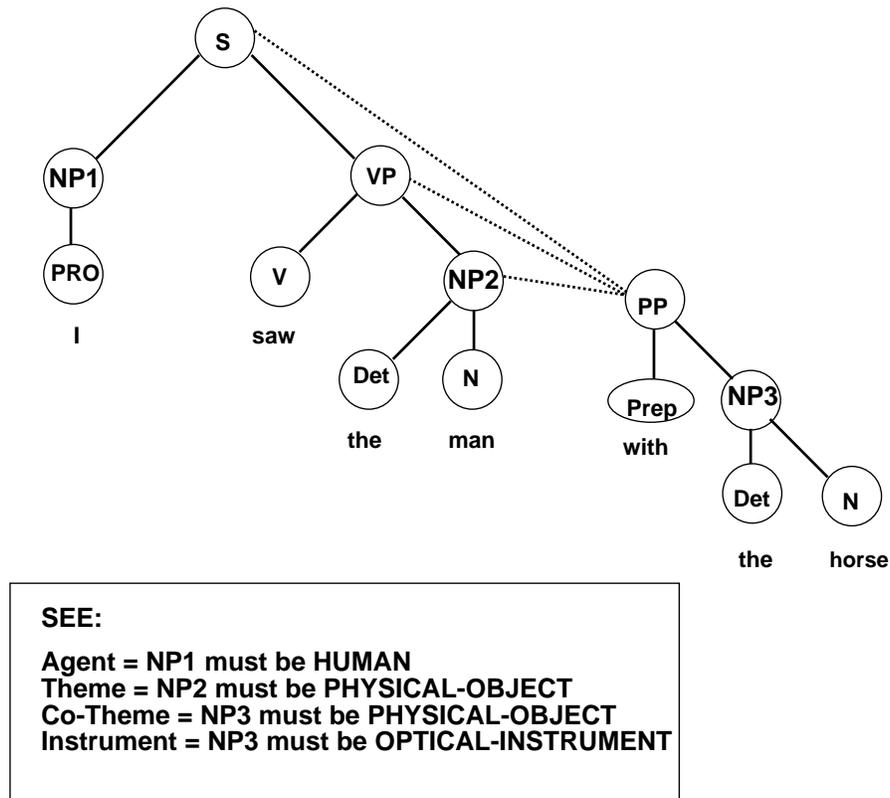
Figure 1: A PP Attachment Ambiguity Showing an Ambiguous Parse Tree and Selectional Constraints on Attachments.

meaningless or less preferred attachment) or the different criteria conflict each other. They are nevertheless highly useful as fall back options for a KB solution when the necessary knowledge is not available or when available knowledge fails to resolve the ambiguity.

3.2   Word Sense Ambiguity

Lexical semantic ambiguity, also known as word sense ambiguity, occurs when a word has more than one meaning (for the same syntactic category of the word). A knowledge-based system selects the meaning that best meets constraints on the composition of the possible meanings with meanings of other words in the text. Selectional constraints on how meanings can be composed with one another are derived from world knowledge, although statistical methods work fairly well in narrow domains where sufficient training data on word senses is available. Typically, the process of checking such constraints is implemented as a search in a semantic network or conceptual ontology. Since such methods tend to be expensive in large-sized networks, constraints are only checked for those pairs of words in the text that are syntactically related to one another.

For example, in the sentence ''The bar was frequented by gangsters,'' the word ''bar'' has a word sense ambiguity: it can mean either an oblong piece of a rigid material or a counter at which liquors or light meals are served. Knowledge of selectional constraints tells us, however, that ''bar'' must be a place of some kind in order for it to be frequented by people. Using such knowledge, a KB-NLP system can correctly resolve the word sense ambiguity and determine that ''bar'' in this case is a liquor-serving place.

3.3   KB Systems for Syntactic and Semantic Analysis

KB-NLP systems for sentence processing can be described in terms of the representations and processing mechanisms they use to apply world knowledge to solve problems such as syntactic and semantic ambiguity. In the following descriptions, we pay particular attention to methods for combining linguistic knowledge with world knowledge. It can be seen that many KB-NLP systems combine the two types of knowledge directly in the representations while others try to combine them during processing.

Early models of sentence processing built in the KB-NLP paradigm attempted integration of multiple types of knowledge in the lexical entries of words. A good example of a model of this kind is the Conceptual Analyzer (CA) model (Birnbaum and Selfridge, 1981) based on Conceptual Dependency representations (Schank and Abelson, 1977). Lexical entries in CA contained rules about what to expect before and after the word in various situations and how to combine the meanings of surrounding words with those of the head words. These lexical entries contained both linguistic and world knowledge. Because of the complete reliance on lexical packaging of all types of knowledge, these models suffered from lack of compositionality and problems of scalability. Small and Rieger (1982) built a model called the Word Expert Parser which used a similar technique. Each word was an expert that knew exactly how to combine its meaning with those of other words in the surrounding context.

Many recent models such as Jurafsky's SAL (1992) use integrated representations of all the different kinds of knowledge in a monolithic knowledge base of integrated constructs called grammatical constructions. Jurafsky's model differs from other integrated models mentioned above in detailing an explicit decomposition of processing into distinct phases called access, selection, and integration of alternative interpretations. This is an example of a decomposition of the language processing task that is orthogonal to standard analyses such as syntax, semantics, and pragmatics.

A related type of integrated model was proposed by Wilks (1975) based on the use of stored conceptual knowledge in the form of templates. Wilks proposed that these templates store constraints on fillers that act as selectional preferences rather than as restrictions. These templates also included syntactic ordering information and as such were packages that combined syntactic, semantic, and conceptual knowledge in an integrated representation.

A different kind of representation is employed in models where syntactic and semantic knowledge are separable but the mappings between them are encoded a priori in individual lexical entries. In this formalism, there are separate syntactic and semantic processes (typically arranged in a syntax-first sequential architecture) but the mapping from syntactic structures to semantic structures is precomputed and enumerated in the lexicon. Models based on lexical semantics, such as the Diana and Mikrokosmos analyzers for machine translation (Meyer, Onyshkevych, and Carlson, 1990; Nirenburg, et al., 1992a;

Onyshkevych and Nirenburg, 1995) are typically sequential models using such representations.

NL-SOAR (Lehman, Lewis, and Newell, 1991; Lewis, 1993a; 1993b) is a model of sentence understanding based on the SOAR architecture for production systems that are capable of learning by chunking (Laird, Newell, and Rosenbloom, 1987). NL-SOAR used architectural constraints from SOAR and a large set of language processing rules to model a range of syntactic phenomena in human sentence processing including structural ambiguity resolution, garden path sentences, and parsing breakdown (such as in center embedded sentences). The model also showed a gradual transition in behavior from deliberative reasoning to immediate recognition. This was made possible by the chunking mechanism in SOAR that produced chunks by combining all the productions that were involved in selecting a particular interpretation for a sentence. A drawback of this approach was the gradual loss of functional independence between the different knowledge sources. Though syntactic and semantic productions were encoded separately to begin with, the chunking process combined them to produce bigger monolithic units applicable to specific types of sentences. Thus, NL-SOAR starts with separate representations of different types of knowledge but gradually builds its own integrated representations. Another drawback of using the SOAR architecture was its serial nature with the consequence that NL-SOAR could pursue only one interpretation of a sentence at any time. Moreover, since productions could contain any type of knowledge, one could encode productions that are rather specific to a particular type of sentence. In fact, chunking would produce such productions that would be applicable in sentences with particular combinations of ambiguities and syntactic and semantic contexts.

Cardie and Lehnert (1991) have extended a conceptual analyzer (such as the CA described earlier) to handle complex syntactic constructs such as embedded clauses. They show that the conceptual parser can correctly interpret the complex syntactic constructs without a separate syntactic grammar or explicit parse tree representations. This is accomplished by a mechanism called lexically-indexed control kernel (LICK) which is essentially a method for dynamically creating a copy of the conceptual parsing mechanism for each embedded clause.

A model called ABSITY that had separate modules for syntax and semantics was developed by Hirst (1988). ABSITY had separate representations of syntactic and semantic knowledge. A syntactic parser similar to PARSIFAL (Marcus, 1980) ran in tandem with a semantic analyzer based on Montague semantics. This model was able to resolve both structural syntactic and lexical semantic ambiguities and produced incremental interpretations. Syntax was provided with semantic feedback so that syntax and semantics could influence each other incrementally. Lexical disambiguation was made possible by the use of individual processes or demons for each word, called <u>Polaroid Words</u>, that interacted with each other to select the meaning that is most appropriate to the context. However, semantic analysis was dependent on the parser producing correct syntactic interpretations. This was a consequence of the requirement of strict correspondence between pieces of syntactic and semantic knowledge for syntax-semantics interaction to work in Hirst's model. Rules for semantic composition had to be paired with corresponding syntactic rules for the tandem design to work.

A noticeable characteristic of Hirst's model was the use of separate mechanisms for solving each subproblem in sentence understanding. The model used a parser based on a capacity limit for syntax, a set of continuously interacting processes through marker passing (e.g., Charniak, 1983, see below) for resolving word sense ambiguities, a ''Semantic Enquiry Desk'' for semantic feedback to syntax, and strict correspondence between syntactic and semantic knowledge for ensuring consistency of incremental

interpretations. This characteristic is inherited completely by a more recent model by McRoy and Hirst (1990) which has more modules than Hirst's original model and appears to be as heterogeneous as its predecessor. This enhanced model is organized in a <u>race-based</u> architecture which simulates syntax and semantics running in parallel by associating time costs with each operation. The model is able to resolve a variety of ambiguities by simply selecting whichever alternative that minimizes the time cost (hence the name ''race-based''). The model employed a Sausage-Machine like two-stage parser for syntactic processing (Frazier and Fodor, 1978) and ABSITY for semantic analysis. The two were put together through an ''Attachment Processor'' and a set of grammar application routines. The attachment processor also consulted three other sets of routines called the grammar consultant routines, knowledge base routines, and argument consultant routines, resulting in a highly complex and heterogeneous model.

COMPERE is a recent model of sentence understanding designed to integrate syntactic and semantic knowledge dynamically during processing (Mahesh, 1995; Mahesh and Eiselt, 1994). COMPERE uses an arbitrating process in a blackboard architecture to choose the best interpretation based on both syntactic and semantic preferences. Additional expert modules can be added to the arbitrating process to account for other linguistic phenomena. A key feature of COMPERE is its close, incremental interactions between syntax and semantics so that most ambiguities can be resolved locally at the earliest possible opportunity in order to minimize the combinatorial effects of multiple ambiguities in a sentence.

Mikrokosmos is a knowledge-based machine translation (KBMT) system that has focused particularly on lexical disambiguation (Onyshkevych and Nirenburg, 1995; Mahesh and Nirenburg, 1995b; Beale, Nirenburg, and Mahesh, 1996). It is one of the largest ever attempts at building a knowledge-based system expressly for NLP. In Mikrokosmos, the meaning of the input text is extracted and represented as instantiated elements of an independently motivated model of the world, that is, an ontology (Carlson and Nirenburg, 1990; Mahesh, 1996). The link between the ontology and the meaning representation is provided by the lexicon, where the meanings of most open class lexical items are defined in terms of their mappings into ontological concepts and their resulting contributions to meaning. The ontology and the lexicon are the two main knowledge sources in the Mikrokosmos system. Information about the nonpropositional components of text meaning such as speech acts, speaker attitudes and intentions, relations among text units, coreferences, etc. is also derived from the lexicon with inputs from other expert ''microtheories.''

The Mikrokosmos core semantic analyzer produces a text meaning representation (TMR) starting from the output of a syntactic analysis module. This knowledge-based engine extracts constraints both from lexical knowledge of a language and from an ontological world model, and applies efficient search and constraint satisfaction algorithms to derive the TMR that best meets all known constraints on the meaning of the given text (Beale et al., 1996; Mahesh and Nirenburg, 1995b). The TMR is intended to serve as an interlingua input for generating the translation in a target language.

In a real-world text, many words have lexical ambiguities. Both the lexicon and the ontology specify constraints on the relationships between the various concepts under consideration, based on language-specific and world knowledge, respectively. The Mikrokosmos analyzer checks each of these selectional constraints by determining how closely a candidate filler meets the constraints on an inter-concept relationship. Closeness is measured by an ontological search algorithm (also known as the <u>semantic affinity checker</u>) that computes a weighted distance between pairs of concepts in the network of concepts present in the ontology. Given a pair of concepts (such as "metal rod" and "place," or,

"drinking place" and "place," to determine which sense of "bar" is closer to a "place" that can be the destination of a physical motion, for instance), the ontological search process finds the least cost path between the two concepts in the ontology where the cost of each link is determined by the type of that link. For example, taxonomic links have a lower cost than "member of" or "has parts" links, and so on. By combining the scores for each link along a path, the core semantic analyzer selects a combination of word senses that minimizes the total deviation from all known constraints and constructs the TMR resulting from the chosen relationships between the concepts (Beale et al., 1996).

Another class of models have used a symbolic equivalent of spreading activation called marker passing in a semantic network to build models of semantic analysis and lexical ambiguity resolution. Among these models, Charniak's (1983) model and Eiselt's ATLAST (Eiselt, 1989) model are particularly interesting. Charniak proposed a semantic processor that used marker passing to initially propose semantic alternatives without being influenced by a syntactic processor (that was running in parallel to the marker passer). There was a third module that combined syntactic preferences with the semantic alternatives to select the combined interpretation. ATLAST (Eiselt, 1989) was a model of lexical semantic and pragmatic ambiguity resolution, as well as error recovery, also using marker passing. ATLAST divided both syntax and semantics into three different modules called the Capsulizer, the Proposer, and the Filter. The Capsulizer was an initial stage that packaged the input sentence into local syntactic units and sent them incrementally to the proposer. The proposer was the marker passer quite akin to the one in Charniak's model. The third module, the filter, combined syntactic and semantic preferences to arrive at the final interpretation. The most important aspect of this model was its ability to recover from its errors without completely reprocessing the sentence by conditionally retaining previously unselected alternatives (Eiselt, 1989). ATLAST was one of the very few models that highlighted the importance of error recovery in shaping the design of a sentence understander. ATLAST's drawback was its limited syntactic capabilities (limited to simple subject-object-verb single-clause sentences) realized in a simple augmented transition-network (ATN) parser (e.g., Allen, 1987) that did not interact in any interesting way with the semantic analyzer.

Models have also been built using connectionist networks with activation and inhibition between nodes being the means of interaction. For example, Waltz and Pollack (1985) built a connectionist model where syntactic and semantic decisions were made in separate networks, but the networks exchanged activation with each other and settled on a combined interpretation at the end. A fundamental problem with these models is the inability of their processing mechanism, spreading activation, to deal with syntactic processing in a large scale. In spite of recent advances in connectionist networks, it is yet to be demonstrated that a connectionist network can process the complex syntax of natural languages and scale up to handle real-world texts.

3.4   Knowledge-Based Solutions to Other NLP Problems

Problems of reference and coreference in NLP often require world knowledge in addition to different types of linguistic knowledge for effective solution. A KB-NLP system can use instances of various concepts in its meaning representation to keep track of references. A unification method may be used to determine coreference between a pair of instances in the meaning representation when suggested by linguistic cues. This method is particularly feasible in a KB-NLP system whose world knowledge

KB has inheritance capabilities. Inherited properties of instances help determine coreference relations whether or not such information is explicitly mentioned in the input text.

Knowledge-based solutions are also applicable to a variety of other problems in NLP such as thematic analysis, topic identification, discourse and context tracking, and temporal reasoning. Knowledge-based methods also play a crucial role in natural language generation for problems such as lexical choice (i.e., selecting a good word or words to realize a given meaning in a language) and text planning (i.e., designing discourse structure, sentence and paragraph boundaries, generating texts with ellipsis and anaphora, and so on, e.g., Viegas and Bouillon, 1994; Viegas and Nirenburg, 1995) .

3.5   Knowledge-Based Inference for Applied NLP

Knowledge-based inference mechanisms for NLP offer partial solutions to a number of practical applications involving text processing. Some of the significant applications include database query, information retrieval and question answering, conceptually-based document retrieval (such as searching the World Wide Web), information extraction, knowledge acquisition from natural-language texts (e.g., Peterson et al., 1994), automatic summarization, knowledge-based machine translation, interpreting non-literal expressions such as metonymy, metaphor, and idioms, document classification, intelligent authoring environments, and intelligent agents that communicate in natural languages. Development of KB-NLP systems for many of these applications is still in somewhat early stages.

4   Research Issues and Discussion

The development of KB-NLP systems both contributes to and benefits from basic and applied research in several areas including computational linguistics, natural-language semantics, knowledge representation, knowledge acquisition, architectures for language processing, and cognitive science. In this section, we briefly examine some of the research issues in knowledge representation and acquisition.

4.1   Knowledge Representation for NLP

KB-NLP requires several types of knowledge, both linguistic (such as grammatical and lexical knowledge) and world knowledge. A key issue in knowledge representation for NLP is how to represent and combine the different types of knowledge. Should there be a single representation that combines all the types of knowledge a priori (perhaps in the form of word experts (Small and Rieger, 1982), in an episodic memory (Riesbeck and Martin, 1986a; 1986b; Schank and Abelson, 1977) or in a ''semantic grammar,'' Burton, 1976) or should there be separate representations that are compatible with one another? The choice is especially important in a multilingual NLP system where it is highly desirable to keep the linguistic knowledge separate for each language but share the common world knowledge across the entire system.

If the representations of linguistic and world knowledge are kept separate, problems arise due

to differences in expressiveness and types of inferences supported by the different representations. While attempts have been made to design uniform representation formats (e.g., predicate-logic based representations like KIF, Genesereth and Fikes, 1992; Gruber, 1993) for all types of knowledge, in practice it seems best to allow slightly different representations and guarantee compatibility across the different representations through an integrated methodology for acquisition (Mahesh, 1996; Mahesh and Nirenburg, 1995a; 1996). Such compatibility can be guaranteed by encoding explicit cross-references (between lexical and world knowledge and between grammatical and lexical knowledge). Such a design also allows one to develop a flexible, blackboard architecture (Mahesh, 1995; Nirenburg and Frederking, 1994; Nirenburg et al., 1994; Reddy, Erman, and Neely, 1973) in which different expert microtheories that solve particular problems in NLP using appropriate knowledge come together to achieve the common goals of KB-NLP (Mahesh and Nirenburg, 1995b; Onyshkevych and Nirenburg, 1995).

The design of a knowledge representation for KB-NLP also affects the accessibility of the represented knowledge for NLP purposes. For example, acquiring lexical entries requires one to find the right concepts in the world knowledge in order to represent meanings of words. Resolving certain word sense ambiguities and interpreting non-literal expressions requires one to measure distances between concepts in the world knowledge. A representation employed for KB-NLP must be designed expressly to support such operations.

4.2 Knowledge Acquisition for NLP

Basic research problems in knowledge acquisition for NLP address possible ways of automating the acquisition of knowledge and the development of methodologies for acquiring linguistic and world knowledge together so that they are compatible with one another.

It is clear that different sources, experts, and constraints govern the acquisition of lexical and world knowledge, especially in a multilingual system. It is important to have an integrated, situated methodology for acquisition where the linguistic and world-knowledge acquisition teams interact with each other regularly and negotiate their choices for each piece of representation. Such an incremental methodology accompanied by a set of tools for supporting the interaction and for quality control is inevitable to guarantee compatibility between the knowledge sources. It will also guarantee that all of the acquired knowledge is in fact usable for KB-NLP purposes. Although it is highly desirable to have general-purpose knowledge bases that can be used for both KB-NLP and other tasks, it has not yet been demonstrated convincingly that knowledge that was not acquired for use in NLP, such as Cyc (Lenat and Guha, 1990), can in fact be used effectively for practical KB-NLP.

In situated knowledge acquisition for KB-NLP, lexical and world knowledge bases are developed concurrently. Often the best choices for each type of knowledge conflict with one another. Practice shows that negotiations to meet the constraints on both a lexical entry and a concept in the ontology leads to the best choice in each case (Mahesh, 1996; Mahesh and Nirenburg, 1995a). It also ensures that every entry in each knowledge base is consistent, compatible with its counterparts, and has a purpose towards the ultimate objectives of KB-NLP. It is also very important to have a well-documented set of guidelines for making choices in both linguistic and world knowledge acquisition (Mahesh, 1996; Mahesh and Nirenburg, 1995a). The ideal method of situated development of knowledge sources for

multilingual NLP is one where an ontology and at least two lexicons for different languages are developed concurrently. This ensures that world knowledge in the ontology is truly language independent and that representational needs of more than one language are taken into account.

Knowledge acquisition for KB-NLP is very expensive in any nontrivial system. The acquisition of world knowledge is typically underconstrained, even in well-defined domains. The situated methodology described above offers a practical way of constraining the amount of world knowledge to be acquired. Only those concepts and conceptual relationships that are required to represent and disambiguate word meanings need to be acquired for KB-NLP purposes. This automatically eliminates certain types of knowledge that may well be within the domain of interest, but will never be used for KB-NLP purposes (e.g., Mahesh, 1996).

In addition to situated development, knowledge acquisition for KB-NLP can be made more tractable by partial automation and by the use of advanced tools. Various attempts have been made to automatically construct grammars and lexicons by analyzing a large corpus of texts. Attempts are also being made to bootstrap the acquisition process so that the KB-NLP system learns and acquires more knowledge as a result of processing input texts. While these are promising approaches to reduce the costs of knowledge acquisition, at present some of the most successful KB-NLP systems have been built by careful manual knowledge acquisition supported by sophisticated tools that help achieve parsimony and high quality in the acquired representations.

5   Summary

Communication in natural languages is possible only when there is a significant amount of general knowledge about the world that is shared among the different participants. An NLP system can make use of such general knowledge of the world as well as specific knowledge of the particular domain in order to solve a range of hard problems in NLP. In this chapter, we have shown how such knowledge can be applied in practice to resolve syntactic and semantic ambiguities and make necessary inferences. We have described algorithms for these solutions, pointed the reader at some of the best implemented systems, and presented some of the current trends and research issues in this field. We would like to emphasize the key practical issue in KB-NLP, namely, that KB solutions are viable and attractive but are often incomplete or rather expensive in practice. It is a good design strategy to evaluate a KB-NLP solution against its alternatives for a particular problem and make the best choice of single method, or, if appropriate, design a multi-engine or human-assisted NLP system. We hope this chapter has helped the reader make a well informed decision.

Defining Terms

Ambiguity: A situation where there is more than one possible interpretation for a part of a text.

Attachment: A syntactic relation between two parts of a sentence where one modifies the meaning of the other.

Blackboard system: A system where several independent modules interact and coordinate with each other by accessing and posting results on a public data structure called the blackboard.

Concept: A unit in world knowledge representation that denotes some object, state, event, or property in the world.

Disambiguation: The process of resolving an ambiguity by selecting one (or a subset) of the possible set of interpretations for a part of a text.

Knowledge: A blanket term for any piece of information that can be applied to solve problems.

Lexical knowledge: Knowledge of words in a natural language. Includes knowledge of syntactic constructs in which the word(s) appear in the language, possible meanings of the words, pronunciation, part of speech, possible inflections, and so on.

Machine translation: Translating a text in one natural language to another natural language by computer.

Ontology: A model of the world that defines each concept that exists in the world as well as taxonomic and other relationships among the concepts. A knowledge base containing information about the domain of interest.

Representation of knowledge: An encoding of knowledge that is computationally tractable.

Selectional constraint: A constraint on the range of concepts with which a concept can have a particular relationship.

Semantics: The branch of linguistics that deals with the meanings of words and texts.

Syntax: The branch of linguistics that deals with the rules that explain the kinds of sentence structure permissible in a language.

Word sense: One of the possible meanings of a word in a language.

World knowledge: Extra-linguistic knowledge. Knowledge of the world or a particular domain that is not specific to any particular natural language.

References

Allen, J. 1987. *Natural Language Understanding*. The Benjamin/Cummings Publishing Company.

Allen, J. 1989. Natural language understanding. section G: Conclusion. In *The Handbook of Artificial Intelligence, Volume IV* ed. A. Barr, P. R. Cohen, and E. A. Feigenbaum, p. 238-239. Addison-Wesley Publishing Company.

Barr, A. and Feigenbaum, E. A. (ed.) 1981. *The handbook of artificial intelligence*. Addison Wesley Publishing Company.

Beale, S., Nirenburg, S., and Mahesh, K. 1996. Hunter-Gatherer: Three Search Techniques Integrated for Natural Language Semantics. In Proceedings of the 13th National Conference on Artificial Intelligence. Portland, Oregon.

Birnbaum, L. and Selfridge, M. 1981. Conceptual analysis of natural language. In *Inside Computer Understanding*, ed. R. Schank and C. Riesbeck, p. 318-353. Lawrence Erlbaum Associates.

Burton, R. 1976. Semantic grammar: An engineering technique for constructing natural language understanding systems. BBN Report No. 3453, Bolt, Beranek, and Newman, Cambridge, MA.

Cardie, C. and Lehnert, W. 1991. A cognitively plausible approach to understanding complex syntax. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, p. 117-124. Morgan Kaufmann.

Carlson, L. and Nirenburg, S. 1990. World Modeling for NLP. Technical Report CMU-CMT-90-121, Center for Machine Translation, Carnegie Mellon University, Pittsburgh, PA.

Charniak, E. 1983. Passing markers: A theory of contextual influence in language comprehension. *Cognitive Science*, 7:171-190.

Cullingford, R. 1978. *Script application: Computer understanding of newspaper stories*. PhD thesis, Yale University, Department of Computer Science, New Haven, CT. Research Report #116.

DeJong, G. 1979. *Skimming stories in real time: An experiment in integrated understanding*. PhD thesis, Yale University, Department of Computer Science, New Haven, CT. Research Report #158.

DeJong, G. 1982. An overview of the FRUMP system. In *Strategies for natural language processing* ed. W. G. Lehnert and M. H. Ringle. Lawrence Erlbaum Associates.

Eiselt, K. P. 1989. *Inference Processing and Error Recovery in Sentence Understanding*. PhD thesis, University of California, Irvine, CA. Tech. Report 89-24.

Frazier, L. and Fodor, J. D. 1978. The Sausage Machine: A new two-stage parsing model. *Cognition*, 6:291-325.

Frederking, R. and S. Nirenburg. 1994. Three Heads Are Better than One. In *Proceedings of Applied Natural Language Processing conference,* ANLP-94. Stuttgart, October.

Frederking, R., Grannes, D., Cousseau, P., and Nirenburg, S. 1993. An MAT Tool and Its Effectiveness. In *Proceedings of the DARPA Human Language Technology Workshop,* Princeton, NJ.

Genesereth, M. and Fikes, R. 1992. Knowledge Interchange Format Version 3.0 Reference Manual. Computer Science Department, Stanford University, Stanford, CA.

Grishman, R. 1986. *Computational linguistics: An introduction.* Cambridge University Press.

Gruber, T. 1993. Toward principles for the design of ontologies used for knowledge sharing. Technical Report KSL 93-04, Stanford Knowledge Systems Lab, August 23, 1993.

Hirst, G. 1988. Semantic interpretation and ambiguity. *Artificial Intelligence*, 34:131-177.

IJCAI Ontology Workshop. 1995. Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, International Joint Conference on Artificial Intelligence, Montreal, Canada, August 1995.

Jurafsky, D. 1992. An on-line computational model of human sentence interpretation. In *Proceedings of the Tenth National Conference on Artificial Intelligence, AAAI 92*, pages 302-308.

Knight, K. and Luk, S. K. 1994. Building a Large-Scale Knowledge Base for Machine Translation. In *Proc. Twelfth National Conf. on Artificial Intelligence,* (AAAI-94).

Laird, J., Newell, A., and Rosenbloom, P. 1987. Soar: An architecture for general intelligence. *Artificial Intelligence*, 33:1-64.

Lebowitz, M. 1983. Memory-based parsing. *Artificial Intelligence*, 21:363-404.

Lehman, J. F., Lewis, R. L., and Newell, A. 1991. Integrating knowledge sources in language comprehension. In *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, pages 461-466.

Lehnert, W. G., Dyer, M. G., Johnson, P. N., Yang, C. J., and Harley, S. 1983. Boris - an experiment in in-depth understanding of narratives. *Artificial Intelligence*, 20(1):15-62.

Lenat, D. B. and Guha, R. V. 1990. *Building Large Knowledge-Based Systems.* Reading, MA: Addison-Wesley.

Lewis, R. L. 1993a. An architecturally-based theory of human sentence comprehension. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, pages 108-113. Lawrence Erlbaum Associates.

Lewis, R. L. 1993b. *An architecturally-based theory of human sentence comprehension.* PhD thesis, Carnegie Mellon University, Computer Science Department, Pittsburgh, PA. Tech. Report. CMU-CS-93-226.

Mahesh, K. 1995. *Syntax-semantics interaction in sentence understanding.* PhD Thesis, College of Computing, Georgia Institute of Technology, Atlanta, GA. Technical Report GIT-CC-95/10.

Mahesh, K. 1996. *Ontology Development: Ideology and Methodology.* Technical Report MCCS-96-292, Computing Research Laboratory, New Mexico State University.

Mahesh, K. and Eiselt, K. 1994. Uniform representations for syntax-semantics arbitration. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, pages 589-594. Hillsdale, NJ: Lawrence Erlbaum.

Mahesh, K. and Nirenburg, S. 1995a. A situated ontology for practical NLP. In *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing,* International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada, August 1995.

Mahesh, K. and Nirenburg, S. 1995b. Semantic Classification for Practical Natural Language Processing. In *Proc. Sixth ASIS SIG/CR Classification Research Workshop: An Interdisciplinary Meeting,* American Society for Information Sciences. October 8, 1995, Chicago IL.

Mahesh, K. and Nirenburg, S. 1996. Meaning Representation for Knowledge Sharing in Practical Machine Translation. In *Proc. FLAIRS-96 special track on Information Interchange,* Florida AI Research Symposium, Key West, FL, May 19-22, 1996.

Marcus, M. 1980. *A Theory of Syntactic Recognition for Natural Language*. MIT Press, Cambridge, MA.

McRoy, S. W. and Hirst, G. 1990. Race-based parsing and syntactic disambiguation. *Cognitive Science*, 14:313-353.

Meyer, I., Onyshkevych, B., and Carlson, L. 1990. Lexicographic principles and design for knowledge-based machine translation. Technical Report CMU-CMT-90-118, Center for Machine Translation, Carnegie Mellon University, Pittsburgh, PA.

Nirenburg, S., Carbonell, J., Tomita, M., and Goodman, K. 1992a. *Machine Translation: A Knowledge-Based Approach*. Morgan Kaufmann Publishers, San Mateo, CA.

Nirenburg, S., Shell, P., Cohen, A., Cousseau, P., Grannes, D., and McNeilly, C. 1992b. The Translator's Workstation. In *Proceedings of the 3rd Conference on Applied Natural Language Processing*. Trento, Italy, April.

Nirenburg, S. and Frederking, R. 1994. Toward Multi-Engine Machine Translation. In *Proceedings of the Human Language Technology Conference,* Princeton, NJ.

Nirenburg, S., Frederking, R., Farwell, D., and Wilks, Y. 1994. Two Types of Adaptive MT Environments. In *Proceedings of the International Conference on Computational Linguistics,* COLING-94, Kyoto, August.

Nirenburg, S. (ed.) 1994. The Pangloss Mark III Machine Translation System. NMSU CRL, USC ISI and CMU CMT Technical Report.

Onyshkevych, B.A. and Nirenburg, S. 1995. A lexicon for knowledge-based MT. *Machine Translation,* 10:1-2, pp. 5-57, Special issue on Building Lexicons for Machine Translation II.

Peterson, J., Mahesh, K., and Goel, A. 1994. Situating Natural Language Understanding within Experience-Based Design. *International Journal of Human-Computer Studies,* Vol. 41, Number 6, Dec. 1994, Pages 881-913.

Ram, A. 1989. *Question-driven understanding: An integrated theory of story understanding, memory and learning*. PhD thesis, Yale University, New Haven, CT. Research Report #710.

Reddy, D., Erman, L., and Neely, R. 1973. A model and a system for machine recognition of speech. *IEEE Transactions on Audio and Electroacoustics*, AU-21:229-238.

Rich, E. and Knight, K. 1991. *Artificial Intelligence.* Second edition. McGraw-Hill, Inc.

Riesbeck, C. K. and Martin, C. E. 1986a. Direct memory access parsing. In *Experience, memory, and reasoning*, ed. J. L. Kolodner and C. K. Riesbeck, p. 209-226. Lawrence Erlbaum, Hillsdale, NJ.

Riesbeck, C. K. and Martin, C. E. 1986b. Towards completely integrated parsing and inferencing. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, pages 381-387. Cognitive Science Society.

Russell, S. J. and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach.* Prentice Hall.

Schank, R. C. and Abelson, R. P. 1977. *Scripts, Plans, Goals, and Understanding*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Schank, R. and Riesbeck, C. K. (ed.) 1981. *Inside Computer Understanding: Five Programs Plus Miniatures.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Small, S. L. and Rieger, C. 1982. Parsing and comprehending with word experts. In *Strategies for natural language processing,* ed. W. G. Lehnert and M. H. Ringle. Lawrence Erlbaum.

Viegas, E. and Bouillon, P. 1994. Semantic Lexicons: the Cornerstone for Lexical Choice in Natural Language Generation. In *Proceedings of the Seventh International Workshop in Natural Language Generation,* Kennebunkport, Maine, June.

Viegas, E. and Nirenburg, S. 1995. The Semantic Recovery of Event Ellipsis: Its Computational Treatment. In *Proceedings of the Workshop on Context in Natural Language Processing,* Fourteenth International Joint Conference on Artificial Intelligence. Montreal.

Waltz, D. L. and Pollack, J. B. 1985. Massively parallel parsing: A strongly interactive model of natural language interpretation. *Cognitive Science*, 9:51-74.

Wilks, Y. 1975. A preferential, pattern-seeking, semantics for natural language inference. *Artificial Intelligence*, 6(1):53-74.

Wilks, Y., Slator, B., and Guthrie, L. 1995. *Electric Words: Dictionaries, Computers and Meanings.* Cambridge, MA: MIT Press.

Further Information

A good introduction to NLP can be found in *Natural Language Understanding* by James Allen or in *Computational Linguistics, An Introduction* by Ralph Grishman. Knowledge-based approaches to NLP are introduced using practical systems in *Inside Computer Understanding: Five Programs Plus Miniatures* by Roger C. Schank and Christopher K. Riesbeck. A useful account of a knowledge-based machine translation system can be found in *Machine Translation: A Knowledge-Based Approach* by Sergei Nirenburg, Jaime Carbonell, Masaru Tomita, and Kenneth Goodman.

The *Handbook of Artificial Intelligence,* edited by Avron Barr and Edward A. Feigenbaum, is also a useful source, especially for early work in the area. The section on *Natural Language Understanding* by James Allen in Volume IV is particularly relevant. Recent AI textbooks such as *Artificial Intelligence* by Elaine Rich and Kevin Knight and *Artificial Intelligence: A Modern Approach* by Stuart Russel and Peter Norvig contain good introductory material and pointers to related work.

Important journals in this area include *Computational Linguistics, Cognitive Science, Machine Translation* and many of the major journals in Artificial Intelligence.

Major associations in the area include the *American Association for Artificial Intelligence (AAAI), Association for Computational Linguistics (ACL),* and the *Cognitive Science Society.* Proceedings of annual conferences of these associations as well as the bi-annual *International Joint Conference on Artificial Intelligence* report some of the latest work in this field.

Useful newsgroups on the Internet include comp.ai.nat-lang, comp.ai.nlang-know-rep, and comp.ai. A compilation of frequently asked questions (FAQs) and answers on NLP is posted regularly on these newsgroups.

Many sites on the World Wide Web contain articles, useful information, and free software relevant to this field. Some useful pointers include the ACL NLP/CL Universe at http://www.cs.columbia.edu/~ radev/cgi-bin/universe.cgi and the Computing Research Laboratory homepage at http://crl.nmsu.edu/Home.html. An electronic archive of papers on computational linguistics and NLP is located at http://xxx.lanl.gov/cmp-lg/