LEXICON DESIGN IN A GERMAN NATURAL LANGUAGE SYSTEM

By

SUSAN ANN MENGEL

Bachelor of Science

Central State University

Edmond, Oklahoma

1982

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 1984

LEXICON DESIGN IN A GERMAN NATURAL
LANGUAGE SYSTEM

Thesis Approved:

G. E. Hedrick
Thesis Adviser

M. J. Folk

Norman N. Durham
Dean of the Graduate College

PREFACE

The design of a German natural language system for computer-assisted instruction is examined in this thesis. Specifically, the beginnings of such a project are considered as well as the component for analyzing the correctness of a German sentence. A prototype that represents this component is implemented to demonstrate the usefulness of such a system in helping a student to learn the German language. The component consists of a German parser and a German lexicon which forms the basis of the system. The German lexicon contains the information about word meanings which enables the parser to recognize a correctly formed sentence and to extract the subject, verb, and object from the sentence.

I wish to thank my committee members, Dr. G. E. Hedrick, Dr. Michael J. Folk, and Dr. Harry Wohlert, for their help and good will while I was writing this thesis. I would like to express my appreciation to Dr. Folk for his patience, guidance, and especially, his class notes which helped me to climb several rungs on the ladder toward being a teacher. I wish to thank Dr. Donald W. Grace for his support and guidance. I also wish to thank Mark Vasoll for helping me with some of the intricacies of the department's computer system.

I wish to express my deepest appreciation to my major advisor, Dr. G. E. Hedrick, for introducing me to natural language processing, for being patient with me during the philosophical arguments about correct grammar, and for helping me with my studies throughout my stay at Oklahoma State University.

Finally, I wish to express my deepest love and affection for my parents and brother who never failed in their support and love for me during my graduate studies in the Department of Computing and Information Sciences.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Motivation

Language is used to express ideas, to learn about other cultures, and to pass on knowledge to future generations. This inherent usefulness has caused it to become the object of study in several areas, such as cognitive psychology, education, and linguistics. These areas analyze language to understand communication, thought processes, and learning capabilities.

Since languages can be quite complex both in structure and meaning, several people in the above areas have used the computer to help to simplify the study of language. Several problems, however, have been encountered in using the computer to process natural languages because of the peculiarities (irregular verbs, compound nouns, words with multiple definitions, etc.) of language and the constraints of the size of memory and processing speed of the computer. Some of the language problems have been solved partially by the use of grammars which define the structure of the language. The size of memory becomes an issue when it is known that the grammar rules should be kept in memory for a fast user response time in a timesharing system. Also, the words and semantic relations between the words should be kept in memory as a lexicon. Expansion of the lexicon, therefore, becomes limited since memory is of a fixed size in some cases and is used partially by the grammar rules. Likewise, the speed and capabilities of the computer's processor or processors is of concern since the natural language system should not be so large that it causes the performance of the computer to degrade.

A solution to the above problems has been offered by Rieger and Small (1981). They have developed a new, but advanced method called "Distributed Word Expert Natural Language Parsing." Their system does not use grammar rules, but instead uses the word itself as a unit of knowledge to determine whether a sentence is grammatically and semantically correct. Each word has its own conceptual relations that indicate its context in a sentence and word senses which give it meaning. A word with its corresponding data is compiled into its own word expert module which decides if the word has meaning in its place in the sentence. The word expert modules formed from a sentence are the only units the parser requires in memory, which solves the space problem. The authors list other advantages of the parser as well; for example, the system allows modular growth since each word expert stands alone, and it could model a person's comprehension of language.

Rieger and Small (1981) have used their system to analyze English text, but they did not mention the use of their system in the area of computer-assisted instruction (CAI). This system would be useful in this area, especially in the field of foreign languages. Generally, Foreign language students are required to write sentences in the language repeatedly in order to gain a natural feel for it. Since computers are helpful in doing repetitive tasks, Rieger and Small's (1981) parser on a computer would be instrumental in analyzing those sentences formed by a student in the chosen language according to both meaning and syntax. The student could gain better and faster mastery of the language in the form of lessons and independent efforts. A study done on a Russian CAI course at Stanford University supports this conclusion since students in the CAI section scored significantly higher on tests than students in the regular section (Suppes, 1981). This conclusion is supported further by Anderson (1980) who reports that the experiments done on memory retention have shown additional study time is needed to protect the memory against the process of forgetting. The need for such a CAI system to analyze sentences of a general nature in a foreign language is even more apparent since most (if not all) systems currently available are ad hoc; i.e. they only address a restricted or narrow subset of the language, and are usually very expensive (Hawkridge, 1983).

Problem to Be Addressed

In view of the need stated above, this paper will concentrate on the construction of a German natural language system in the area of CAI with a particular emphasis on the design of the lexicon.

Method and Limitations

Since this project is of a large size, this paper only concentrates on the beginnings of the overall system with further development left to future research. The system includes a parser similar to the Distributed Word Expert Parser of Rieger and Small (1981) and a lexicon containing a subset of words from the German language with their appropriate conceptual and semantic information. This subset demonstrates the projected usefulness of the system in determining whether a sentence is grammatically correct even when it contains words with multiple definitions.

The scope of this system is limited to selected portions from the first four chapters of Crean et al (1981) which is the textbook used in the beginning German courses at Oklahoma State University. These portions include the nominative and accusative cases, the present tense of verbs, predicate nominatives, predicate adjectives, nouns, compound nouns, adverbs, certain prepositions, personal pronouns, statements, and questions. Sentences which the system can analyze must have at least a subject and a verb. Sentences may not contain conjunctions, numbers, adjectives other than predicate adjectives, or prepositions other than "in," "aus," and "nach" which must take a noun indicating a physical location for their object. Further limitations and sentence syntax are discussed in Chapter III.

Definitions

To ensure the clarity of this paper, the following definitions are provided:

Parse - To apply the grammar, conceptual rules, or semantic rules of a natural language system to a sentence so that the structure of the sentence may be analyzed. It also may be the final result of such an analysis.

Lexicon - That structure which contains the selected words of a natural language system, their meanings, and their conceptual dependencies.

Ambiguous Sentence - A sentence with more than one valid parse.

Lexical Analysis - The identification of words in the language.

Syntax Analysis - The analysis of the sentence construction.

Semantic Analysis - The analysis of the meaning of words.

Sense - The semantic content of a word including its part of speech, gender, case, verb case, number, and meanings. It also may refer only to the meaning of the word.

Organization of Study

Chapter I deals with the motivation for the system and the method of construction. Chapter II includes a literature review to discuss papers in the following areas: syntax analysis, lexicon design, parts of speech, and CAI. Chapter III discusses the implementation considerations of the German language computer representation, German parser, and German lexicon. The system operation and performance on the computer is shown in Chapter IV. Chapter V presents the evaluation of the system, conclusions of the study, and suggestions for future research.

CHAPTER II

LITERATURE REVIEW

Syntax Analysis

Through many years of research, several methods have been developed to put language on the computer. Some of these methods incorporate the use of grammars so that the role of syntax (the structure of a sentence) is emphasized more than the role of semantics (the meaning of the words in the sentence).

Transformational Grammar is one method that utilizes a grammar and that relies heavily upon syntax. The transformational rules consist of the following: the structural description to define the general form of the grammar rule and the structural changes which may be performed on the structural description. For example, the word "up" occurs in two different places in the following similar sentences:

John will pick up the blocks.

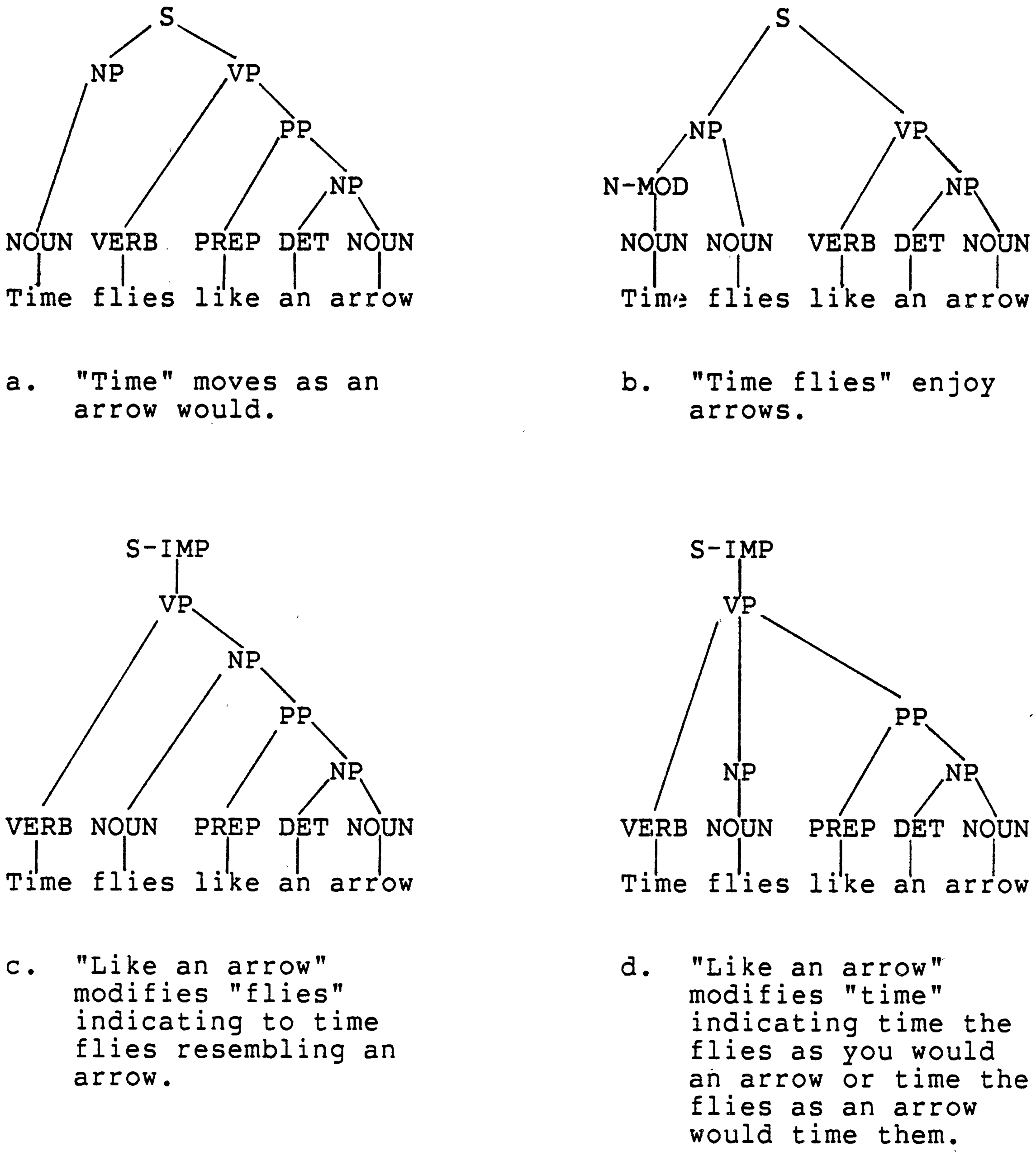
John will pick the blocks up.

A transformational rule to accommodate both sentences might look like:

```
          Particle Movement
    SD:  X verb particle nounphrase Y
         1 - 2 -    3    -    4    - 5
    SC:  1 - 2 -    0    -   4+3   - 5
```

where X and Y are components irrelevant to the transformation, SD is the structural description, SC is the structural change, and the 4+3 in the SC indicates the particle may come after the noun phrase (Akmajian et al, 1979). A transformational grammar system uses these transformational rules to determine all possible parses of a sentence. Sometimes a sentence may have several valid grammatical parses and as a result, becomes ambiguous. The sentence "Time flies like an arrow" may have four separate parses and five separate interpretations as illustrated in figure one. The reason for this ambiguity is evident since "time," "flies," and "like" are classified as more than one part of speech. The original model of transformational grammar could not choose one parse as being the "correct parse" since it relied only on syntax (Wilks, 1975). Also, Transformational Grammar has been found to be unamenable to computer applications in natural language processing due to theoretical difficulties in determining its computability (Wilks, 1975).

Another method which uses a grammar is the augmented transition network (ATN). Once again, this method relies heavily upon syntax, but is able to do some minor semantic analysis. An ATN is a form of an augmented pushdown automaton. It has the state and stack information along with a set of register contents and the ability to perform



a. "Time" moves as an arrow would.

b. "Time flies" enjoy arrows.

c. "Like an arrow" modifies "flies" indicating to time flies resembling an arrow.

d. "Like an arrow" modifies "time" indicating time the flies as you would an arrow or time the flies as an arrow would time them.

KEY:

S = Sentence	S-IMP = Imperative Sentence
NP = Noun Phrase	VP = Verb phrase
N-MOD = Noun Modifier	PP = Prepositional Phrase
DET = Determiner	PREP = Preposition

Figure 1. Parse trees and interpretations of the sentence "Time flies like an arrow."

arbitrary computational tests and actions associated with the state transitions (Woods, 1980). ATN's are able to capture linguistic regularities, but are tied closely to their application making them nonportable and nonextensible. Cascaded ATN's which are a set of ATN's that work concurrently have been developed to allow more semantic analysis during parsing (Woods, 1980). For example, one ATN can do lexical analysis, another syntax analysis, and another semantic analysis. Cascaded ATN's, however, would still be tied closely to their application.

Since understanding was not usually stressed as much as syntax in systems that used grammar, other researchers began to look for more syntax- and language-independent methods of natural language processing. Among these people was Schank (1975) who developed the Conceptual Dependency Theory which states that language has a conceptual base that is interlingual. His overall goal was language understanding and so he created a conceptual analyzer to parse a sentence into its conceptual components rather than its grammatical components. In this way, his system, MARGIE, could paraphrase and make inferences about a sentence (See figure two). Schank's conceptual rules enabled MARGIE to make these paraphrases and inferences as illustrated in figure three. Schank also included in the system the way a human being might interpret a sentence. For instance, the sentence "Time flies like an arrow" would be interpreted probably as a metaphor as in part a of figure one rather

```
INFERENCE MODE

INPUT:     John told Mary that Bill wants a book.
OUTPUT1:   A book about what?
OUTPUT2:   Mary knows that Bill wants a book.
OUTPUT3:   Bill wants to come to have a book.
OUTPUT4:   Bill wants someone to cease to have a book.
OUTPUT5:   Bill wants to read a book.


PARAPHRASE MODE

INPUT:     John killed Mary by choking Mary.
OUTPUT1:   John strangled Mary.
OUTPUT2:   John choked Mary and she died because she could
           not breathe.
OUTPUT3:   Mary died because she was unable to inhale some
           air and she was unable to inhale some air because
           John grabbed her neck.
```

Figure 2. Sample output from MARGIE (Schank, 1975, pp. 2,3).

```
                           ---> man
     p          o       R |
man <=> ATRANS <- book <--|
                          |
                           ---< someone
```

KEY: p = past
<=> = relationship between subject and verb
ATRANS = action primitive indicating a transfer of ownership
o = object
R = recipient

Figure 3. Conceptual structure of the sentence, "The man took the book" (Schank, 1975, p. 28).

than any of the four other ways mentioned in the rest of figure one. Schank argued that a natural language system should accept that same interpretation in order to approximate human language understanding. For all its abilities, Schank found MARGIE inadequate as a human language understanding model since it lacked the knowledge of the context in which the sentence was given. Because of this deficiency, MARGIE made some irrelevant inferences. Schank remedied much of this problem by developing other systems based on his conceptual dependency theory which analyze related text rather than single sentences.

While Schank was developing his Conceptual Dependency Theory, Wilks (1975a) was working on his Preference Semantics System to translate English into French. This system, like Schank's, did not utilize a grammar as such, but did use a well-defined semantic structure among the words. Wilks' system could analyze sentences as well as small paragraphs of text and could handle words with multiple meanings. His method worked by breaking a sentence into fragments (phrase, clause, or primitive sentence). Each fragment would then be split into formulas, one for each word, that would be combined into semantic templates with agent, action, and object as the major components. The sentence "John gave Mary the book" would initially have two templates: one for "John gave Mary" and another for "John gave book." The system would try to expand the first template by attempting to determine the indirect object as defined by its semantic rule. Finding "Mary," the system would reject the first template since "Mary" is already the direct object and proceed to the second template. The system would expand the second template successfully and give the following semantic representation:

```
John <-> gave <-> book
          ^        ^
          |        |
         Mary     the
```

where <-> denotes the nonpreferential link between the formulas and -> denotes the preferential dependency established (Wilks, 1975a). Wilks' system also handled the problem of tying pronouns and their antecedents together by using the semantic structure information associated with the words and sentence. His system could determine that the "them" in the sentence "The soldiers fired at the women and we saw several of them fall" referred to the "women" by using a common sense rule that "if an animate object strikes another animate object, the second one is more likely to fall than the first one." In this way, Wilks' Preference Semantics System could tell the difference between feminine and masculine pronouns so the French translation could be effected.

Building upon the work of these two people, Rieger and Small (1981) developed their Distributed Word Expert Natural Language Parser. They wanted their parser to be able to deal with multiple word meanings and also to work at a conceptual level to allow word disambiguation to be aided with open-ended world knowledge. They also wanted their parser to deal first with the "irregularities" of language claiming that the "regularities" would be handled as a natural side effect. They took this approach since many methods deal first with the "regularities" making these methods unable to deal with the entire structure of language. Their system accomplished its tasks by allowing each word in a sentence to be compiled into a word expert; i.e., a program which can identify a word's meaning by using the word's semantic structure, by asking questions of other word experts, and by checking the conceptual information of the text being parsed. In analyzing "the deep pit," the Word Expert Parser would compile the "the" expert and allow it to execute. The "the" expert would decide that it is an article (begins a picture or noun phrase construction) and wouÌˆld terminate. The "deep" expert would then run and be unable to determine its meaning. The "pit" expert would run and be unable to determine its meaning since it can be either a "fruit pit" or a "hole in the ground." The "deep" expert would reawaken and constrain "pit" to be a "hole in the ground" or a "person." The "pit" expert would take that information and decide that its meaning is a "hole in the ground" and would terminate. The "deep" expert would terminate last with a meaning of "large volume" (Rieger and Small, 1981).

Lexicon Design

Since the syntax analysis methods are so diverse, each uses its own particular lexicon to hold the words and semantic dependencies among the words. As a result, a definition for the universal structure of a lexicon does not seem to exist in the current literature. Cercone (1978), however, has done extensive work on the design of a lexicon and the representation of word meanings. He advocates the use of morphological analysis, such as separating affixes from words and separating the components of compounds. He asserts that the use of this method can save storage since only the root forms would have to be stored, can provide interpretive assistance of a word by analyzing the affixes, can help to learn the meaning of new words by a preliminary analysis of the structure of the sentence and the affix relationships, and can supply the meanings of words having affixes without having to store the affixes (such as "non" which means negation). He, like Schank and Wilks, has devised his own method of defining lexical entries and word meanings and so his method is closely tied to his parser as their methods are. Instead of using semantic primitives to describe the meanings of sets of words like Schank and Wilks, Cercone allows each word to express its own concepts asserting that each word has its own particular senses of cause, motion, time, etc. He (1983) also has done some work on minimal perfect hash function search to retrieve information from the lexicon faster than a binary or tree search could retrieve it, but only has achieved a low collision rate with small sets of words. With that in mind, he has suggested splitting the lexicon into two or more hierarchies with the most frequently used words referenced first.

Parts of Speech

Apart from determining what part of speech a word represents in a sentence, subject-verb agreement, and tense of the verb, is the problem presented by the use of compound nouns (complex nominals). Jones (1983) makes the point that it is impossible to put all compounds in the lexicon. This condition is true since compounds may be formed at will. For instance, if a loaf of bread were on a table, that table might be called the "bread table" merely to distinguish it from another table. "Bread table" may have several meanings, such as "a table with bread on it" or "a table for bread." The natural language system should be able to determine the correct meaning even if only "bread" and "table" are in the lexicon separately. This problem has been addressed by several people, such as Levi, Lees, and Li, but none have been successful in analyzing all compound nouns (Downing, 1977).

The use of pronouns has caused some problems as well in the area of determining the antecedent of a pronoun. Wilks (1975b) has tackled the problem by separating pronouns into three types: type A which are resolved by the conceptual content of the words, type B which are resolved by analytic inferences, and type C which are resolved by weak generalizations about the course of events in the world. The sentence "Give the bananas to the monkeys although they are not ripe, because they are very hungry" contains pronouns of type A. The first "they" is attributed to "bananas" since being ripe is usually a characteristic of plants and the second "they" is assigned to "monkeys" since being hungry is generally a characteristic of animate beings. When a pronoun reference cannot be resolved by semantic dependencies, the system changes to the extended mode of inference to handle types B and C. A sentence with type B pronouns is "John drank the whiskey from the glass, and it felt warm in his stomach." The system needs extra help in determining the antecedent of "it" since it could be the "glass" or the "whiskey." To resolve this difficulty, the system uses an inference rule specifying "what is in a part of X is in X." The semantic description "drink" indicates that liquid is taken to the inside of an animate being and so the "whiskey" is determined to be inside of John. "It" is also inside of John by virtue of being in his stomach and, therefore, "it" and "whiskey" are linked together. Type C pronouns require even more help in determining their referents. The sentence "The dogs chased the cats, and I heard one of them squeal with pain" contains a type C pronoun, "one." The system uses a "common sense inference rule" specifying "animate beings that are pursued by animate beings may be unpleasantly affected" to match "one" to "cat." This method of resolving anaphora (pronoun references) helps especially in the area of machine translation where a neuter pronoun in English may translate into a masculine or feminine pronoun in some other language.

Computer-Assisted Instruction

CAI is a field of immense potential in a learning environment, but has not been applied to its fullest extent. The reasons for this lack of utilization are due in part to hardware costs, hardware restraints and failures, lack of quality courseware, expensive development costs of courseware, and lack of teacher involvement. The situation is illustrated further by Amarel (Wilkinson, 1983):

> From a survey conducted by the National Center of Educational Statistics (Goor, Melmud, & Ferris, 1981) of a national sample of public school districts, a picture of fairly broad, but extremely shallow, penetration emerges. Although about half of the school districts report having at least one microcomputer (micro), only 3% of the districts have 20 or more micros available for instructional use. Translated into availability to schools in the "have" districts, less than 3% of primary schools and less than 1% of secondary schools have 20 or more terminals or micros for student use. The most frequently reported use of on-line instruction is in computer literacy courses, which typically provide some familiarity with a programming language. Remedial and compensatory education, on the one hand, and providing additional challenge for high-achieving students, on the other, round out the types of student activities reported. As of 1981, then, CAI (computer-aided instruction) was not used to deliver main-line instruction or even to provide a significant addition to traditional curricular offerings, and it was least used by the middle range of students (p. 20).

The impact of teachers on a CAI curriculum was illustrated by the PLATO Elementary Mathematics and Reading Demonstration done in the mid-1970's (Amarel in Wilkinson, 1983). With this system, teachers had the responsibilities of controlling access to terminals and integrating the computerized lessons with ongoing instruction. When hardware failures occurred, schedules were disrupted and teachers had to reshuffle student assignments. Also, teachers had to spend time ensuring that all students took their turn at the terminals. With all of the teachers' time required on PLATO, this study showed that teachers are as involved as the students in CAI and should be considered as much as the students are in the design of a CAI system so they will have more of a desire to use it.

Hawkridge (1983) further expands on some of the problems with CAI. He states that many people in education are completely opposed to CAI for the following reasons: high quality software and courseware will not be available in sufficient quantity and variety, CAI will lead to overreliance on mediated learning (learning through media) rather than enactive learning (learning through direct experience), and teachers will have an unwillingness and inability to deal with CAI. These problems are certainly real, but they can be tackled. The first problem is in part due to misjudging the scope of the content of a course, due to unknowingly making the content unrelated to the method of instruction, due to rapidly expanding fields causing the content to be outdated quickly, and due to not taking advantage of the full capacities of the computer medium. With feedback and further study in system utilization, student progress, and instruction usefulness, these deficiencies can be remedied. The second problem is not a problem in the view of some researchers. They assert computers can be designed to communicate languages and mathematics naturally and effectively to students, thus helping to make these subjects easy instead of difficult. The solution to the third problem can be aided by involving teachers more in the design of CAI packages and giving teachers additional training in the capabilities of computers.

Despite these problems, researchers remain optimistic about CAI and its future. CAI should help to develop new instructional paradigms, to open new methods of expression to handicapped people, and to provide more individualized instruction to students. CAI systems designed to allow remedial students more instruction than advanced students, to allow adequate feedback on a student's progress, to adapt easily to new course requirements, and to incorporate a teacher's instructional method, serve to accomplish these goals.

Two systems which include these traits do so through the use of artificial intelligence. One is the system of Weischedel, Voge, and James (1978) and the other is the system, MALT (Machine Language Tutor), designed by Koffman

(1975). The system of Weischedel et al is a small German tutor used to teach a first course by presenting reading passages in German to develop reading comprehension skills and the ability to compose well-formed answers to questions about the passages. The system is able to analyze a student's answer both syntactically and semantically so an erroneous answer may be analyzed and the specific error transmitted back to the student. This immediate feedback allows the student a better chance to remember and avoid subsequent errors. The second system, MALT, is designed for an introductory course in computer science as an aid to the teaching of machine language. It is a generative system which means it both generates and solves meaningful problems so that students may be given problems to challenge them and to strengthen their weaknesses. MALT accomplishes these tasks by keeping track of the student's performance and by monitoring a student's solution to a particular problem. A problem is posed from a framework of acceptable propositions with certain variables randomly generated by MALT which also computes the solution(s). The student is given an outline of how to solve the problem in the form of subtasks. The program of the beginning student can be monitored one line at a time to provide immediate feedback in case of errors and the student can be given MALT's solution as an additional instructional aid. As the student progresses, less monitoring is used and more difficult problems are posed so individualized instruction is achieved. These two systems serve to indicate the direction CAI development is taking.

Summary

This chapter discussed several methods used in natural language processing to show how semantic based methods, such as the Conceptual Dependency Theory and Preference Semantics, developed from syntactic-based methods, such as Transformational Grammar and augmented transition networks. The semantic-based methods led to the development of the Word Expert Parser. These methods all used a lexicon of some type that could be designed with morphological analysis performed upon the words to conserve storage. Retrieval methods for these words included tree searches, binary searches, and minimal perfect hash function searches. Some additional considerations for a natural language system were handling complex nominals and anaphora. Also, the evaluation of a natural language system designed for CAI could be accomplished by using feedback of student progress, system utilization, and instructional usefulness. A well-constructed CAI system allowed individualized student instruction, adapted easily to new course requirements, and incorporated the teacher's instructional method.

CHAPTER III

IMPLEMENTATION CONSIDERATIONS

In order to keep the design of this system at a manageable level, its scope is limited only to selected portions from the first four chapters of Crean et al (1981) as mentioned in Chapter I. Simplicity is desirable in the early stages of design since natural language systems can become quite complex even when restricting the set of syntax and semantics in the language to be handled.

German Language Computer Representation

The German alphabet is made up of the letters "a" through "z" along with the additional vowels, "ä," "ö," and "ü," and the additional letter, "ess-tset," which represents a double "s." Since many computer terminals do not have the facilities to represent these special letters, they are represented by their English equivalents, "ae," "oe," "ue," and "ss," respectively.

German Parser

One of the considerations in the design of the system is the definition of the syntax accepted by the parser. The sentence forms in figure four illustrate this syntax.

Subject Verb Object.

Verb Subject Object?

Ja, Subject Verb Object.

Nein, Subject Verb Object.

Element Verb Subject Object.

Interrogative-Adverb-Phrase Verb Subject?

Interrogative-Pronoun-Phrase Verb Subject?

Where:

Subject := Noun Phrase

Object := Noun Phrase or Predicate Adjective

Verb := Present Tense Verb

Element := Adverb or Prepositional Phrase

Interrogative-Adverb-Phrase :=

```
   Interrogative Adverb
or Interrogative Adverb, Noun
or Interrogative Adverb, Following Word, Noun
```

Interrogative-Pronoun-Phrase :=

```
   Interrogative Pronoun or
or Interrogative Pronoun, Following Word, Noun
```

Noun Phrase := Noun or Pronoun or Article, Noun

Prepositional Phrase := Preposition, Noun

Following Word := Word that forms part of a phrase

Figure 4. Syntax forms accepted by the German parser.

Variations on these forms are accepted as well; for example, adverbs may come after the verb and at the end of a sentence, prepositional phrases may come at the end of a sentence, and objects do not have to be present. The subject and verb, however, may not be separated and the verb must be the second element in the sentence unless it is at the beginning of a sentence that forms a question.

Another of the considerations in the design of this system is the human element. The ultimate goal of this project is to complete a well designed CAI system to aid in teaching students the German language. The beginning student often finds it difficult in the initial stages of learning a language to set aside the grammar rules of his or her own native language. As a result, the sentence that a student forms in the new language may contain the words of the language, but may be in the syntax order of the native language. For instance, in the English language the word, "not," is used within the verb to negate the meaning of a sentence. In German, "nicht" is used for the same purpose, but it must come before the phrase it negates or at the end of the sentence to negate the entire sentence. This subtle difference between the positioning of the two words may go unnoticed by the student at first. The sentence "Er nicht ist interessant" may seem quite natural to the student even if it is grammatically incorrect since the subject and verb are separated. On a given input from a student, the word order, therefore, may not be entirely correct. To accommodate this type of error, the parser should not impose such a strict syntax upon the student's input so that processing stops at the word that is out of order (since other errors may be present as well). The parser should reflect the error and continue processing so the student may receive the maximum benefit from the error correction done by the system. For this reason, the parser in this system accepts any word order as long as the elements themselves are not split apart (i.e., an article and preposition are followed by a noun and other phrases are not separated). When a word is out of order, its condition is denoted by an appropriate error message as reflected in the sample runs in Chapter IV.

Since the parser does not depend strictly upon word order to analyze a sentence, meanings must be given to the words to enable the parser to distinguish among the functions of the words. For instance, in the two sentences:

(1) Wie viele Flugzeuge fliegen die Amerikaner?
(2) Wie viele Flugzeuge fliegen?

"Flugzeuge" serves as a noun, but it is the object in sentence one and the subject in sentence two. The choice between subject and object in sentence one is difficult because both nouns are plural, but it may be resolved by having the verb in the sentence take an "animate being" for a subject and a "thing" for an object. The subject in sentence two defaults to "Flugzeuge" since it is the only noun in the sentence, but the verb also needs to be able to take a "thing" for a subject and nothing for an object. If a semantic match cannot be effected to resolve the meaning of the sentence, the parser attempts a match based upon the word order.

Since the parser is constructed to work using the semantic content of the words, only the semantic content of the words is used. This condition is imposed so that the parser does not have to rely on the correctness of the student's input. For instance, The German language grammar contains many "markers" which help in determining the meaning and function of a word. Nouns must be capitalized and the articles must reflect the case of the nouns they modify. It is not safe to assume that the beginning student is able to apply these grammar rules correctly every time, especially since this system is designed to help the student correct such errors.

German Lexicon Design

Without the lexicon, the parser cannot determine what German words are valid or what meaning and function the German words may have. The lexicon, therefore, is an important part of the natural language system. It should be designed so that the parser may extract the words and senses easily.

With the above feature in mind, the lexicon for this system is designed to consist of two parts: the valid German words and the corresponding senses. The words are stored in alphabetical order in memory so that a binary search can be used to locate a word from a sentence (a binary search is fast enough for the small set of the German language implemented in this system). No morphological analysis is performed upon the words so that nothing may hinder the binary search, except the location of the words themselves. Each word entry contains the word, the number of senses, and a pointer to the first sense in the sense entry table. The senses may then be referenced one after the other from the first sense until the number of senses is exhausted. A sense entry contains:

(1) the part of speech,
(2) the gender,
(3) the case,
(4) the verb case,
(5) the number,
(6) the senses,
(7) the following word.

The following word is used to indicate that a preposition may be in the same sentence with a particular verb and to indicate that a word follows another word to form a phrase (for instance, "viele" follows "wie" to form the adverbial interrogative phrase "wie viele"). Table I lists the valid entries for the first five semantic fields of a sense, Table II lists the valid semantic primitives used in the sense fields, and figure five contains some sample lexical entries. Even though it is not indicated in Table I, the fields, gender, case, verb case, and number, may take a "NOTAPPLICABLE" entry of "z" since these fields may not apply to all parts of speech. The primitives used to define the meanings of words are broad in definition, but they are

TABLE I

FIRST FIVE VALID SEMANTIC FIELD VALUES

Field	Name	Symbol	Description
Parts of Speech:	ARTICLE	t	article adjective
	PREP	r	preposition
	PRONOUN	p	pronoun
	ADJECTIVE	j	predicate adjective
	VERB	v	present tense verb
	ADVERB	d	adverb
	NOUN	n	noun
Genders:	MASCULINE	m	masculine gender
	FEMININE	f	feminine gender
	NEUTER	n	neuter gender
Cases:	NOMINATIVE	n	nominative case
	ACCUSATIVE	a	accusative case
	FOLLOW	f	used to indicate that the word may only follow another word
Verb Cases:	FIRST	f	first person
	SECOND	s	second person
	THIRD	t	third person
	POLITEFORM	p	polite you
Number:	SINGULAR	s	singular
	PLURAL	p	plural

TABLE II

VALID SEMANTIC SENSE FIELD PRIMITIVES

Field	Name	Symbol	Description
Senses:	NULLSENSE	0	null sense
	PICPRODUCER	1	picture producer (introduces noun phrase)
	BE	2	existence
	THINK	3	mental transfer
	KIND	4	predicate adjective
	INTERROGATIVE	5	interrogative word
	NEGATIVE	6	German word "nicht"
	POSITIVE	7	German word "gern"
	DO	8	act
	ANIMATE	9	animate being
	WORLD	10	anything
	NIL	11	no object
	SAME	12	predicate noun
	PLACE	13	physical location
	THING	14	physical object
	LANGUAGE	15	spoken word
	HUMAN	16	human being
	BEAST	17	animal
	AFFIRM	18	German word "ja"
	DISSENT	19	German word "nein"
	TRAVEL	20	movement

Lexical Word and Sense Structure

```
          word  number-of-senses  first-sense-pointer

 part   gender case verb number  sense   sense  sense   follow
  of                case           1       2      3      word
speech                          meaning subject object



                           die 4 11

                        t f n z s   1   0   0
                        t f n z p   1   0   0
                        t f a z s   1   0   0
                        t f a z p   1   0   0



                           ist 3 25

                      v z z t s   2  10   4  in
                      v z z t s   2  10  12  in
                      v z z t s   2  10  11  in



                           was 4 52

                     p z n t s  14   5   0  fuer
                     p z n t p  14   5   0  fuer
                     p z a z p  14   5   0  fuer
                     p z a z p  14   5   0  fuer



                          nicht 1 33

                     d z z z z   6   0   0  wahr
```

Figure 5. Sample lexical entries.

precise enough to enable the parser to determine the correctness of a sentence formed from the system's subset of the German language. Each word may contain different senses, but it must be classified as only one part of speech so that the relative simplicity of the parser may be maintained. Attempting to distinguish among different parts of speech of one word leads to the problems discussed in Chapter II with the sentence "Time flies like an arrow."

Summary

In this chapter, the basic design of the overall system is considered. The scope of the system is defined to encompass selected portions of the first four chapters from a beginning German textbook. The extra letters in the German alphabet are given a representation that can be typed on most computer terminals. The parser for the system is designed to accept any word order as long as phrases are not separated and to utilize the lexicon to aid in distinguishing among the functions of the words. The lexicon is constructed to contain the valid German words and semantic meanings and to facilitate word and sense extraction by the parser.

CHAPTER IV

THE SYSTEM

The system is coded in the C language and consists of a total of 34 modules. The maintenance and user's manual may be found in the appendixes, but the actual programs are on tape and may be obtained from the Computing and Information Sciences Department of Oklahoma State University. Since the system is of a large size and rather complex, the algorithms are not presented in this paper, but are included in the programs on the tape. The computer on which the system runs is a Perkin-Elmer 3230.

Operation

When a user submits a German sentence, the system reads the sentence one word at a time checking to see that each word contains valid characters from the German alphabet and to see that each word is in the lexicon. If a sentence is entered successfully (i.e. no invalid characters are present and all words are in the lexicon), control of the system passes to the sentence analysis routines. These routines determine the correctness of the syntax and, to a limited extent, the semantics of the sentence.

The analysis routines first initialize the senses of the words as completely as possible from the lexicon. If a word has several senses that contain the same information in some of the sense fields, that information is stored with the word in memory. If some of the fields are different, a marker indicating "unknown" is placed in the appropriate field. Initializing each word's sense is necessary so that the parse of the sentence may be expedited. During this procedure of initialization, the system checks to make sure that no more than one verb is present so that processing may continue.

After the initialization process terminates, the system begins to analyze the sentence one word at a time in word order on the basis of the word's part of speech. Each part of speech has its own module which processes the word sent to it and then calls the appropriate module for the next word in the sentence. When an article is found, the article module checks for a noun phrase by determining that a noun is next in the word sequence. When an adjective is found, the adjective module makes sure that the word is a predicate adjective by checking for the presence of an object in the sentence. If no object is found, the predicate adjective becomes the object in the sentence and processing continues. When an adverb is found, the adverb module checks for the position of the word to make sure that it comes at the beginning of the sentence or after the verb. Certain adverbs need to be at the beginning of the sentence; for example, "ja," "nein," and interrogative adverbs. Other adverbs need to come after the verb, such as "nicht" and "gern." These positions are verified and "nicht," "gern," and interrogative adverbs are checked for more than one occurrence. Some of the interrogative adverbs may form a phrase which may or may not have an object (noun). If an object is present, the adverb module sets a pointer to the interrogative adverb and passes it to the noun module. The preposition module determines if the preposition is allowed to be in the sentence by checking the verb's following word field. If the preposition is present in that field, the routine then verifies the presence of a prepositional phrase by determining that a noun follows next in the word sequence. If a noun is present, a pointer to the preposition is passed to the noun module. The pronoun module checks for interrogative pronouns as well as personal pronouns. If the interrogative pronoun forms a phrase, an object must be present, such as in "Was für Tische haben Sie?" where "Tische" is the object of the interrogative phrase. The module determines if a phrase is formed by checking the following word field for the presence of a word and then the next word in the sentence against the following word. If a match results, the routine then verifies that the next word in the sentence is a noun and passes a pointer that indicates the position of the interrogative pronoun to the noun module. If a phrase is determined not to be present, the pronoun routine assigns the pronoun to be the subject or object of the sentence. The noun module first determines if the word is part of a phrase by checking the previous pointer for a nonnegative number which points to the first word in the phrase. If the previous word is a preposition, the noun module calls another routine to link the preposition and object together; otherwise, it assigns the noun to be the subject or object in the sentence. The verb module verifies that the verb is at least the second element in the sentence. After the end of the sentence has been encountered and control has been passed back to the verb module, it checks for the presence of a subject and that the subject and verb are together in the sentence. The verb module then passes control to the module which controls the linkage of the words in the sentence.

In order to link the words of the sentence together, the system first attempts to link the subject and object to the verb. Every subject, verb, and object sense is used in every combination possible. If one combination results in a denser match than another, those senses involved in the match are saved and the previous senses are discarded. The density of the subject and verb link is based on the case, verb case, number, and meaning of the subject, and the verb case, number, and meaning of the subject sense of the verb. The density of the object's link to the subject and verb is based on the current verb sense being analyzed. If the verb does not take an object, the link is based on whether an object is present. If the verb takes a predicate adjective for an object, the link is based on the meaning of the object. If the verb takes a predicate noun for an object, the subject and object are compared according to gender, case, verb case, number, and meaning. If the verb takes a direct object, the link is based on the object's meaning. If only partial or no linkage results, the system attempts a linkage by letting the object be the subject and the subject be the object. If some linkage is accomplished, the system then endeavors to link the articles and interrogative words to their objects by using the same method of trying all possible combinations of senses.

The system also looks for proper punctuation. For instance, the German words, "ja" and "nein," must be followed by a comma and questions need to be followed by question marks. In the case of nouns, the system checks for capitalization.

Sample Runs

The system has been designed to effect a small trace of the analysis process to show the senses of the words before the process begins and after the process terminates. The three sentences:

(1) Er ist amerikaner nicht wahr.
(2) Die Flugzeuge gern fliegen die Amerikaner.
(3) Ja, wir sind gern in Frankreich.

are used as examples to demonstrate the execution of the system. In the print-out of the traces which show the sentence words with their semantic content, the character,

"%," indicates an unresolved semantic field and the number, "-1," indicates an unresolved meaning in the word's sense fields. The message preceded by the three equal signs is the input prompt, the messages preceded by three asterisks are system messages to the user, and messages preceded by the letter, "t," are the trace messages. In the trace output, each word and its sense are listed along with five additional fields: "punc" which indicates the punctuation immediately following the word, "link" which indicates the position of the word in the lexicon, "complete" which indicates that a word only has one sense in the lexicon, "capitalize" which contains a nonzero number if the word is capitalized, and "prev_ptr" which contains a nonnegative number if the word must be linked to another word, such as a noun that must be linked to an article.

The trace of the first sentence is shown in Tables III and IV. In Table III, the sentence is read into the system and the sentence words are initialized as completely as possible. This initialization is reflected in the trace. The words, "er," "nicht," and "wahr," are completely initialized since they have only one sense entry in the lexicon. The word, "ist," is not completely initialized since it may have a predicate noun or a predicate adjective for an object or no object at all. The word, "amerikaner," is not completely initialized since it may be masculine or feminine and singular or plural. The system then tries every combination of senses of the subject, verb, and object

TABLE III

PART I OF FIRST SAMPLE RUN

```
===Input German Sentence:
Er ist amerikaner nicht wahr.

*** Analysis of sentence beginning.

ttt subject_ptr = -1  object_ptr = -1  verb_ptr = -1

ttt The sentence words and senses:
   er                              part_of_speech =   p  gender     =   m
      punc =       complete =   1  verbcase       =   t  wcase      =   n
      link =   33  prev_ptr =  -1  sing_or_plural =   s  capitalize =   1
                   sense 0  =   9     sense 1  =   0     sense 2  =   0
      follow =

   ist                             part_of_speech =   v  gender     =   z
      punc =       complete =   0  verbcase       =   t  wcase      =   z
      link =   69  prev_ptr =  -1  sing_or_plural =   s  capitalize =   0
                   sense 0  =   2     sense 1  =  10     sense 2  =  -1
      follow = in

   amerikaner                      part_of_speech =   n  gender     =   %
      punc =       complete =   0  verbcase       =   t  wcase      =   z
      link =    1  prev_ptr =  -1  sing_or_plural =   %  capitalize =   0
                   sense 0  =  16     sense 1  =   0     sense 2  =   0
      follow =

   nicht                           part_of_speech =   d  gender     =   z
      punc =       complete =   1  verbcase       =   z  wcase      =   z
      link =   93  prev_ptr =  -1  sing_or_plural =   z  capitalize =   0
                   sense 0  =   6     sense 1  =   0     sense 2  =   0
      follow = wahr

   wahr                            part_of_speech =   j  gender     =   z
      punc =    .  complete =   1  verbcase       =   z  wcase      =   z
      link =  135  prev_ptr =  -1  sing_or_plural =   z  capitalize =   0
                   sense 0  =   4     sense 1  =   0     sense 2  =   0
      follow =
```

TABLE IV

PART II OF FIRST SAMPLE RUN

```
----------------------------------------------------------------------

*** The noun, "amerikaner," should begin with a capital letter in the sentence.
*** "nicht wahr" needs to be preceded by a comma.
*** This sentence needs to end with a question mark.

*** Please note the indicated errors and make corrections.

ttt subject_ptr = 0  object_ptr = 2  verb_ptr = 1

ttt The sentence words and senses:
  er                                part_of_speech =  p  gender      =  m
     punc =         complete =   1  verbcase       =  t  wcase       =  n
     link =    33   prev_ptr =  -1  sing_or_plural =  s  capitalize  =  1
                    sense 0  =   9     sense 1  =   0    sense 2  =   0
     follow =

  ist                               part_of_speech =  v  gender      =  z
     punc =         complete =   0  verbcase       =  t  wcase       =  z
     link =    69   prev_ptr =  -1  sing_or_plural =  s  capitalize  =  0
                    sense 0  =   2     sense 1  =  10    sense 2  =  12
     follow = in

  amerikaner                        part_of_speech =  n  gender      =  m
     punc =         complete =   0  verbcase       =  t  wcase       =  n
     link =     1   prev_ptr =  -1  sing_or_plural =  s  capitalize  =  0
                    sense 0  =  16     sense 1  =   0    sense 2  =   0
     follow =

  nicht                             part_of_speech =  d  gender      =  z
     punc =         complete =   1  verbcase       =  z  wcase       =  z
     link =    93   prev_ptr =  -1  sing_or_plural =  z  capitalize  =  0
                    sense 0  =   6     sense 1  =   0    sense 2  =   0
     follow = wahr

  wahr                              part_of_speech =  j  gender      =  z
     punc =     .   complete =   1  verbcase       =  z  wcase       =  z
     link =   135   prev_ptr =  -1  sing_or_plural =  z  capitalize  =  0
                    sense 0  =   4     sense 1  =   0    sense 2  =   0
     follow =

----------------------------------------------------------------------
```

which are "er," "ist," and "amerikaner," respectively, to link the sentence together. Table IV contains the analysis of the sentence and the output of the words and senses after the analysis has terminated. This sentence is not without errors as reflected by the system messages on punctuation and noun capitalization, but even with those errors the system is able to resolve the meaning of the sentence. The words, "er" and "amerikaner," are determined to be in the nominative case since the verb takes a predicate noun for an object and "amerikaner" is resolved to be singular and masculine since it is a predicate noun describing "er." The presence of "nicht wahr" indicates that the sentence is a question which is shown in the sentence analysis messages. The correct rendering of the sentence is, therefore, "Er ist Amerikaner, nicht wahr?" as the system messages indicate.

The trace of the second sentence is shown in Tables V and VI and is of interest since either noun could be the subject due to the fact that "Flugzeuge" is plural and "Amerikaner" can be plural. The meaning of the sentence is that Americans enjoy flying airplanes rather than airplanes enjoy flying Americans. As shown in Table V, the verb, "fliegen," does not have resolved meanings for its subject and object. "Fliegen" may take a "thing" as a subject with no object, an "animate being" as a subject with no object, or an "animate being" as a subject with a "thing" as an object. After trying all the senses, the system does determine that "Amerikaner" is the subject and "Flugzeuge"

TABLE V

PART I OF SECOND SAMPLE RUN

```
------------------------------------------------------------------------------

===Input German Sentence:
Die Flugzeuge gern fliegen die Amerikaner.

*** Analysis of sentence beginning.

ttt subject_ptr = -1  object_ptr = -1  verb_ptr = -1

ttt The sentence words and senses:
   die                             part_of_speech =   t  gender     =   f
      punc =       complete =   0  verbcase       =   z  wcase      =   %
      link =   28  prev_ptr =  -1  sing_or_plural =   %  capitalize =   1
                   sense 0  =   1     sense 1  =   0     sense 2  =   0
      follow =

   flugzeuge                       part_of_speech =   n  gender     =   f
      punc =       complete =   1  verbcase       =   t  wcase      =   z
      link =   46  prev_ptr =  -1  sing_or_plural =   p  capitalize =   1
                   sense 0  =  14     sense 1  =   0     sense 2  =   0
      follow =

   gern                            part_of_speech =   d  gender     =   z
      punc =       complete =   1  verbcase       =   z  wcase      =   z
      link =   55  prev_ptr =  -1  sing_or_plural =   z  capitalize =   0
                   sense 0  =   7     sense 1  =   0     sense 2  =   0
      follow =

   fliegen                         part_of_speech =   v  gender     =   z
      punc =       complete =   0  verbcase       =   %  wcase      =   z
      link =   42  prev_ptr =  -1  sing_or_plural =   %  capitalize =   0
                   sense 0  =  20     sense 1  =  -1     sense 2  =  -1
      follow =

   die                             part_of_speech =   t  gender     =   f
      punc =       complete =   0  verbcase       =   z  wcase      =   %
      link =   28  prev_ptr =  -1  sing_or_plural =   %  capitalize =   0
                   sense 0  =   1     sense 1  =   0     sense 2  =   0
      follow =

   amerikaner                      part_of_speech =   n  gender     =   %
      punc =    .  complete =   0  verbcase       =   t  wcase      =   z
      link =    1  prev_ptr =  -1  sing_or_plural =   %  capitalize =   1
                   sense 0  =  16     sense 1  =   0     sense 2  =   0
      follow =

------------------------------------------------------------------------------
```

TABLE VI

PART II OF SECOND SAMPLE RUN

```
*** "gern" should come after the verb.
*** The verb should be the second element in the sentence.

*** Please note the indicated errors and make corrections.

ttt subject_ptr = 5  object_ptr = 1  verb_ptr = 3

ttt The sentence words and senses:
   die                             part_of_speech =   t  gender     =   f
      punc =       complete =   0  verbcase       =   z  wcase      =   a
      link =   28  prev_ptr =  -1  sing_or_plural =   p  capitalize =   1
                   sense 0  =   1     sense 1  =   0     sense 2  =   0
      follow =

   flugzeuge                       part_of_speech =   n  gender     =   f
      punc =       complete =   1  verbcase       =   t  wcase      =   a
      link =   46  prev_ptr =   0  sing_or_plural =   p  capitalize =   1
                   sense 0  =  14     sense 1  =   0     sense 2  =   0
      follow =

   gern                            part_of_speech =   d  gender     =   z
      punc =       complete =   1  verbcase       =   z  wcase      =   z
      link =   55  prev_ptr =  -1  sing_or_plural =   z  capitalize =   0
                   sense 0  =   7     sense 1  =   0     sense 2  =   0
      follow =

   fliegen                         part_of_speech =   v  gender     =   z
      punc =       complete =   0  verbcase       =   t  wcase      =   z
      link =   42  prev_ptr =  -1  sing_or_plural =   p  capitalize =   0
                   sense 0  =  20     sense 1  =   9     sense 2  =  14
      follow =

   die                             part_of_speech =   t  gender     =   f
      punc =       complete =   0  verbcase       =   z  wcase      =   n
      link =   28  prev_ptr =  -1  sing_or_plural =   p  capitalize =   0
                   sense 0  =   1     sense 1  =   0     sense 2  =   0
      follow =

   amerikaner                      part_of_speech =   n  gender     =   f
      punc =    .  complete =   0  verbcase       =   t  wcase      =   n
      link =    1  prev_ptr =   4  sing_or_plural =   p  capitalize =   1
                   sense 0  =  16     sense 1  =   0     sense 2  =   0
      follow =
```

is the object as shown in Table VI.

The trace of the third sentence is shown in Tables VII and VIII. This particular sentence has no errors in it and simply reflects a successfully processed sentence by the fact that the subject and verb are reflected in the system messages as well as the absence of an object.

System Performance

Since the system is relatively small, the analysis of a sentence does not take more than a few seconds if that long. The only performance degradation occurs when the lexicon is read into memory.

Summary

The system operation along with some sample runs has been presented in this chapter. The system performs its analysis upon a sentence one word at a time and then links the words together by considering every combination of the senses of the words. The sample runs show that the system can recognize errors and resolve the meanings of words. The system performance is found not to be poor because of its quick response time due to its small size.

TABLE VII

PART I OF THIRD SAMPLE RUN

```
===Input German Sentence:
Ja, wir sind gern in Frankreich.

*** Analysis of sentence beginning.

ttt subject_ptr = -1  object_ptr = -1  verb_ptr = -1

ttt The sentence words and senses:
   ja                              part_of_speech =   d gender      =   z
      punc =    ,  complete =   1  verbcase       =   z wcase       =   z
      link =   70  prev_ptr =  -1  sing_or_plural =   z capitalize  =   1
                   sense 0 =  18     sense 1 =   0     sense 2 =   0
      follow =

   wir                             part_of_speech =   p gender      =   z
      punc =       complete =   1  verbcase       =   f wcase       =   n
      link =  143  prev_ptr =  -1  sing_or_plural =   p capitalize  =   0
                   sense 0 =   9     sense 1 =   0     sense 2 =   0
      follow =

   sind                            part_of_speech =   v gender      =   z
      punc =       complete =   0  verbcase       =   % wcase       =   z
      link =  117  prev_ptr =  -1  sing_or_plural =   % capitalize  =   0
                   sense 0 =   2     sense 1 =  10     sense 2 =  -1
      follow = in

   gern                            part_of_speech =   d gender      =   z
      punc =       complete =   1  verbcase       =   z wcase       =   z
      link =   55  prev_ptr =  -1  sing_or_plural =   z capitalize  =   0
                   sense 0 =   7     sense 1 =   0     sense 2 =   0
      follow =

   in                              part_of_speech =   r gender      =   z
      punc =       complete =   1  verbcase       =   z wcase       =   z
      link =   65  prev_ptr =  -1  sing_or_plural =   z capitalize  =   0
                   sense 0 =  13     sense 1 =   0     sense 2 =   0
      follow =

   frankreich                      part_of_speech =   n gender      =   n
      punc =    .  complete =   1  verbcase       =   t wcase       =   z
      link =   47  prev_ptr =  -1  sing_or_plural =   s capitalize  =   1
                   sense 0 =  13     sense 1 =   0     sense 2 =   0
      follow =
```

TABLE VIII

PART II OF THIRD SAMPLE RUN

```
*** Subject:  wir
*** No object
*** Verb:  sind

ttt subject_ptr = 1  object_ptr = -1  verb_ptr = 2

ttt The sentence words and senses:
   ja                                part_of_speech =   d  gender      =   z
      punc =     ,  complete =   1   verbcase       =   z  wcase       =   z
      link =    70  prev_ptr =  -1   sing_or_plural =   z  capitalize  =   1
                    sense 0  =  18      sense 1  =   0     sense 2  =   0
      follow =

   wir                               part_of_speech =   p  gender      =   z
      punc =        complete =   1   verbcase       =   f  wcase       =   n
      link =   143  prev_ptr =  -1   sing_or_plural =   p  capitalize  =   0
                    sense 0  =   9      sense 1  =   0     sense 2  =   0
      follow =

   sind                              part_of_speech =   v  gender      =   z
      punc =        complete =   0   verbcase       =   f  wcase       =   z
      link =   117  prev_ptr =  -1   sing_or_plural =   p  capitalize  =   0
                    sense 0  =   2      sense 1  =  10     sense 2  =  11
      follow = in

   gern                              part_of_speech =   d  gender      =   z
      punc =        complete =   1   verbcase       =   z  wcase       =   z
      link =    55  prev_ptr =  -1   sing_or_plural =   z  capitalize  =   0
                    sense 0  =   7      sense 1  =   0     sense 2  =   0
      follow =

   in                                part_of_speech =   r  gender      =   z
      punc =        complete =   1   verbcase       =   z  wcase       =   z
      link =    65  prev_ptr =  -1   sing_or_plural =   z  capitalize  =   0
                    sense 0  =  13      sense 1  =   0     sense 2  =   0
      follow =

   frankreich                        part_of_speech =   n  gender      =   n
      punc =     .  complete =   1   verbcase       =   t  wcase       =   z
      link =    47  prev_ptr =   4   sing_or_plural =   s  capitalize  =   1
                    sense 0  =  13      sense 1  =   0     sense 2  =   0
      follow =
```

CHAPTER V

EVALUATION, CONCLUSIONS, AND SUGGESTED FUTURE RESEARCH

Evaluation

One of the things that the system cannot handle very well is the presence of the polite pronoun, "Sie." Since the German word, "sie," can mean "she," "they," and "you" (if "sie" is capitalized, but this system does not differentiate between capitalized and noncapitalized words) both in the nominative and accusative cases, the system can resolve the meaning of "sie" to the polite "you" only in a sentence with a singular predicate noun, such as "Sie sind Student." Also since the semantic primitives used to define the senses are fairly broad in definition, it is possible for the system to accept "nonsensical" sentences, such as "Der Tisch ist freundlich" which means that the table is friendly.

The system, however, has a quick response time which is desirable in an interactive system. It also is flexible enough to resolve the meaning of the sentence as well as to account for the proper syntax the sentence should have even when the sentence does not have the proper word order. This trait is important for a CAI system so that most errors a student might make can be corrected to achieve the goal of helping the student learn the language.

Conclusions

This system obviously is not ready to be used for CAI, but it does reflect the general direction in which such a system might be designed; i.e. to handle as many language errors as possible. The acquisition of a new language is not always easy for a student and so the more practice a student gains the better his or her chances are for learning the language. The system presented in this thesis can handle some of the repetitious practice a student needs in learning a new language.

Suggested Future Research

One of the things that can be done to improve this system involves narrowing the scope of the semantic primitives by defining more primitives and allowing more than one primitive to define a word. For instance, "freundlich" can be defined to be a modifier of an animate being by using the semantic primitives, KIND and ANIMATE, to prevent it from being linked to inanimate objects. Enlarging the set of primitives facilitates machine translation. For example, if a translation of the user's input is desired, the semantic primitives can help to choose the English words with the same meaning for the proper English translation.

Another area of improvement to the system occurs with the addition of morphological analysis. Since the system currently does not use morphological analysis of any kind, it cannot accept any compound words in German that are not in the lexicon. Nouns frequently are combined to form compounds in the German language which may not be in a German dictionary. Morphological analysis is needed to analyze these words so that their meanings may be inferred. Morphological analysis has the added advantage of reducing the size of the lexicon by eliminating the need to store all forms of a word. This process can extract the stem of a word so that only the stem needs to be stored in the lexicon as in the case of a verb. A routine to check spelling can be used during morphological analysis to correct misspelled words so that the user does not have to submit the sentence again. The addition of morphological analysis can degrade the response time of the system, but as the lexicon starts to approach the size of memory, the price may be worth the extra processing time so that a larger subset of the German language may be implemented.

The search for a word in the lexicon can be improved by implementing a hash function with a low collision rate. Also, as the lexicon becomes larger, it might be useful to split it into several parts according to the frequency of word use. The most frequently used words can be searched first so that the majority of the search time is spent on words that usually occur in a sentence.

In order to be able to recognize the meanings of words more fully, the system can be improved to analyze bodies of text. The context of the surrounding text material may facilitate the process of resolving a word's meaning. For instance, pronouns and their antecedents can be linked together as discussed in Chapter II. Also, when the system is extended to process words with different parts of speech, the surrounding text material again may help to clarify the meanings of words.

Lessons in the German language can improve and can be constructed around the system so specific user responses may be defined and that definition used in resolving the meaning of the sentence. These lessons also can be personalized to monitor a particular student's progress so that more advanced students may be challenged with more difficult material. Properly designed lessons, as discussed in Chapter II, serve to enhance this project and help it to achieve the long-term goal of facilitating the instruction of the German language and perhaps, eventually, other languages.

SELECTED BIBLIOGRAPHY

Akmajian, Adrian, Richard A. Demers, and Robert M. Harnish. Linguistics: An Introduction to Language and Communication. The MIT Press, Cambridge, Massachusetts, 1979.

Amarel, Marianne. "The Classroom: An Instructional Setting for Teachers, Students, and the Computer." Classroom Computers and Cognitive Science. Alex Cherry Wilkinson, ed. Academic Press, Inc., New York, 1983, 15-29.

Anderson, John R. Cognitive Psychology and Its Implications. W. H. Freeman and Company, San Francisco, 1980.

Cercone, Nick. "Morphological Analysis and Lexicon Design for Natural-Language Processing." Computers and the Humanities, 11 (1978), 235-238.

Cercone, Nick, Max Krause, and John Boates. "Minimal and Almost Minimal Perfect Hash Function Search with Application to Natural Language Lexicon Design." Computers and Mathematics with Applications, 9, 1 (1983), 215-231.

Crean, Jr., John E., Claude Hill, Franz Langhammer, and Kenneth Wilcox. Deutsche Sprache und Landeskunde. Random House, Inc., New York, 1981.

Downing, Pamela. "On the Creation and Use of English Compound Nouns." Language, 53 (1977), 810-842.

Hawkridge, David. New Information Technology in Education. The Johns Hopkins University Press, Baltimore, Maryland, 1983.

Jones, Karen Sparck. "So What about Parsing Compound Nouns." Automatic Natural Language Parsing. Karen Sparck Jones and Yorick Wilks, eds. Ellis Horwood Limited, Chichester, 1983, 164-168.

Jones, Karen Sparck and Yorick Wilks, eds. Automatic Natural Language Parsing. Ellis Horwood Limited, Chichester, 1983.

Keenan, Edward L., ed. *Formal Semantics of Natural Language*. Cambridge University Press, London, 1975.

Lehnert, Wendy G. and Martin H. Ringle, eds. *Strategies for Natural Language Processing*. Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey, 1982.

Rieger, Chuck and Steve Small. "Toward a Theory of Distributed Word Expert Natural Language Parsing." *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11, 1 (Jan. 1981), 43-51.

Schank, Roger C. "Conceptual Dependency: A Theory of Natural Language Understanding." *Cognitive Psychology*, 3 (1972), 552-631.

Schank, Roger C. *Conceptual Information Processing*. North-Holland Publishing Company, Amsterdam, 1975.

Schank, Roger C. and Kenneth M. Colby, eds. *Computer Models of Thought and Language*. W. H. Freeman and Company, San Francisco, 1973.

Schank, Roger C., Michael Lebowitz, and Lawrence Birnbaum. "An Integrated Understander." *American Journal of Computational Linguistics*, 6, 1 (Jan.-March 1980), 13-30.

Schank, Roger C. and Christopher K. Riesbeck, eds. *Inside Computer Understanding: Five Programs Plus Miniatures*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1981.

Shapiro, Stuart C. "Generalized Augmented Transition Network Grammars for Generation from Semantic Networks." *American Journal of Computational Linguistics*, 8, 1 (Jan.-March 1982), 12-25.

Small, Steve and Chuck Rieger. "Parsing and Comprehending with Word Experts (A Theory and Its Realization)." *Strategies for Natural Language Processing*. Wendy G. Lehnert and Martin H. Ringle, eds. Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey, 1982, 89-147.

Sowa, John F. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1984.

Suppes, Patrick, ed. *University-Level Computer-Assisted Instruction at Stanford: 1968-1980*. Institute for Mathematical Studies in the Social Sciences, Stanford, California, 1981.

Venezky, Richard L. "Evaluating Computer-Assisted Instruction on Its Own Terms." *Classroom Computers and Cognitive Science*. Alex Cherry Wilkinson, ed. Academic Press, Inc., New York, 1983, 31-49.

Waltz, David L. "The State-of-the-Art in Natural-Language Understanding." *Strategies for Natural Language Processing*. Wendy G. Lehnert and Martin H. Ringle, eds. Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey, 1982, 3-32.

Weischedel, Ralph M., Wilfried M. Voge, and Mark James. "An Artificial Intelligence Approach to Language Instruction." *Artificial Intelligence*, 10 (1978), 225-240.

Wilkinson, Alex Cherry, ed. *Classroom Computers and Cognitive Science*. Academic Press, Inc., New York, 1983.

Wilks, Yorick Alexander. *Grammar, Meaning and the Machine Analysis of Language*. Routledge & Kegan Paul Ltd., London, 1972.

Wilks, Yorick. "An Intelligent Analyzer and Understander of English." *Communications of the ACM*, 18, 5 (May 1975a), 264-274.

Wilks, Yorick. "A Preferential, Pattern-Seeking, Semantics for Natural Language Inference." *Artificial Intelligence*, 6 (1975b), 53-74.

Wilks, Yorick. "Preference Semantics." *Formal Semantics of Natural Language*. Edward L. Keenan, ed. Cambridge University Press, London, 1975c.

Woods, William A. "Cascaded ATN Grammars." *American Journal of Computational Linguistics*, 6, 1 (Jan.-March 1980), 1-12.

APPENDIXES

APPENDIX A

USER'S GUIDE

SUSAN MENGEL
USER'S GUIDE FOR CAI
JULY 23, 1984

PURPOSE: Cai is a German natural language system designed to be useful for computer-assisted instruction.

INPUT: Cai accepts as input German sentences submitted by the user. German words have the following character sequences within them to represent the additional letters in the German alphabet:

```
ae - a umlaut  oe - o umlaut  ue - u umlaut
ss - ess-tset.
```

The words that may be used in these sentences can be obtained by consulting the linklex system manuals located in /grad/sam/thesis/german/link.

The sentences must be in the following forms as found in the beginning German textbook by Crean et al:

```
          Subject Verb Object.

          Verb Subject Object?

        Ja, Subject Verb Object.

       Nein, Subject Verb Object.

      Element Verb Subject Object.

Interrogative-Adverb-Phrase Verb Subject?

Interrogative-Pronoun-Phrase Verb Subject?
```

Where:

```
   Subject := Noun Phrase

   Object := Noun Phrase or Predicate Adjective

   Verb := Present Tense Verb

   Element := Adverb or Prepositional Phrase

   Interrogative-Adverb-Phrase :=
           Interrogative Adverb
        or Interrogative Adverb, Noun
        or Interrogative Adverb, Following Word,
             Noun
```

```
Interrogative-Pronoun-Phrase :=
        Interrogative Pronoun
     or Interrogative Pronoun, Following Word,
          Noun

Noun Phrase := Noun or Pronoun or Article, Noun

Prepositional Phrase := Preposition, Noun

Following Word := Word that forms part of a
                    phrase.
```

OUTPUT: Cai outputs on a successful analysis, the subject, verb, and object of the sentence. On an unsuccessful parse, cai outputs the errors present in the sentence.

LIMITATIONS: Cai is designed only to handle selected material taken from the first four chapters of Crean et al.

SAMPLE RUN: Cai may be found in /grad/sam/thesis/german. To exit the system, simply return after the input prompt has been given.

```
%cai
===Input German Sentence:
Ja, er ist alt.

*** Subject:  er
*** Object:  alt
*** Verb:  ist

===Input German Sentence:
Wie viele Flugzeuge fliegen die amerikaner.

*** The noun, "amerikaner," should begin with a capital
    letter in the sentence.
*** This sentence needs to end with a question mark.

*** Please note the indicated errors and make corrections.

===Input German Sentence:

%
```

CONTACT: Should any problems arise during the execution of cai, contact Susan Mengel in the Computing and Information Sciences Department at Oklahoma State University.

REFERENCE:

Crean, Jr., John E., Claude Hill, Franz Langhammer, and Kenneth Wilcox. Deutsche Sprache and Landeskunde. Random House, Inc., New York, 1981.

ADDITIONAL INFORMATION:

Further documentation about the system may be found in the programs themselves, in the linklex system which maintains the lexicon located in /grad/sam/thesis/german/link, and in this thesis located in /grad/sam/thesis.

APPENDIX B

MAINTENANCE GUIDE

SUSAN MENGEL
MAINTENANCE MANUAL FOR CAI
JULY 19, 1984

PURPOSE: Cai is a German natural language system designed to be useful for computer-assisted instruction.

INPUT: Cai accepts as input:

1. A file, "lexicon," containing the valid German words and senses for the system. The file is in the following form:

```
number of words  number of senses
senses:  part of speech  gender  case
         verb case  number  sense1  sense2
         sense3  follow
words:   word  number of senses
         first sense pointer
```

   where the senses are in corresponding word order and the words are in alphabetical order.

```
Example:     4 5
             j z z z z 4 0 0 %
             n m z t s 16 0 0 %
             n f z t p 16 0 0 %
             n f z t s 16 0 0 %
             n f z t p 16 0 0 %
             alt 1 0
             amerikaner 2 1
             amerikanerin 1 3
             amerikanerinnen 1 4
```

   The senses and words are read into two separate arrays, lex_senses and lex_words, respectively.

2. German sentences submitted by the user. German words have the following character sequences within them to represent the additional letters of the German alphabet:

   ae - a umlaut oe - o umlaut ue - u umlaut
   ss - ess-tset.

OUTPUT: Cai outputs appropriate error messages should errors occur, such as when the lexicon cannot be read properly. Cai outputs an analysis of the German sentence submitted by the user.

If the cai programs are compiled with the -DDEBUG qualifier, cai outputs several debug messages concerning the entrance and exit of modules.

If the cai programs are compiled with the -DTRACE qualifier, cai outputs a trace of the sentence analysis in the form of words and corresponding senses.

OUTLINE: Individual outlines of modules are contained within the programs themselves, but the overall outline is as follows:

```
1.  Input lexicon
2.  While (user inputs German sentence)
        read sentence words
        find words in the lexicon
        initialize senses of the words
        analyze the sentence by extracting the
          subject, verb, and object
        link the words together
        output the results
3.  Endwhile
4.  End
```

MODULES: Main Program

driver.c - Drives the German natural language system

Subprograms

1. adjectiv.c - Drives adjective analysis
2. adverb.c - Drives adverb analysis
3. art.c - Drives article analysis
4. asgnsens.c - Assigns subject, verb, and object senses to the appropriate sentence words
5. checkend.c - Determines correct end punctuation
6. clrinp.c - Clears input buffer
7. comsens.c - Compares German word senses
8. endpunc.c - Checks for end punctuation
9. getsent.c - Reads a user's sentence
10. getword.c - Reads a German word
11. initlex.c - Reads the lexicon into memory

12. initsens.c - Initializes the senses of the sentence words

13. linkart.c - Links an article to its noun

14. linkint.c - Links an interrogative word to its object

15. linkprep.c - Links a preposition to its object

16. linksbob.c - Links the subject and object

17. matart.c - Matches the "best" article sense to the noun sense

18. matobj.c - Matches the "best" object sense to the sentence

19. matsub.c - Matches the "best" subject sense to the verb sense

20. noun.c - Drives noun analysis

21. prep.c - Drives preposition analysis

22. procsent.c - Drives the sentence analysis routines

23. pronoun.c - Drives the pronoun analysis

24. retrieve.c - Retrieves a word from the lexicon

25. tieart.c - Indicates density of article and noun link

26. tieobj.c - Indicates density of object and sentence link

27. tiesent.c - Ties the sentence together

28. tiesubj.c - Indicates density of subject and verb link

29. valpunc.c - Checks for valid punctuation marks

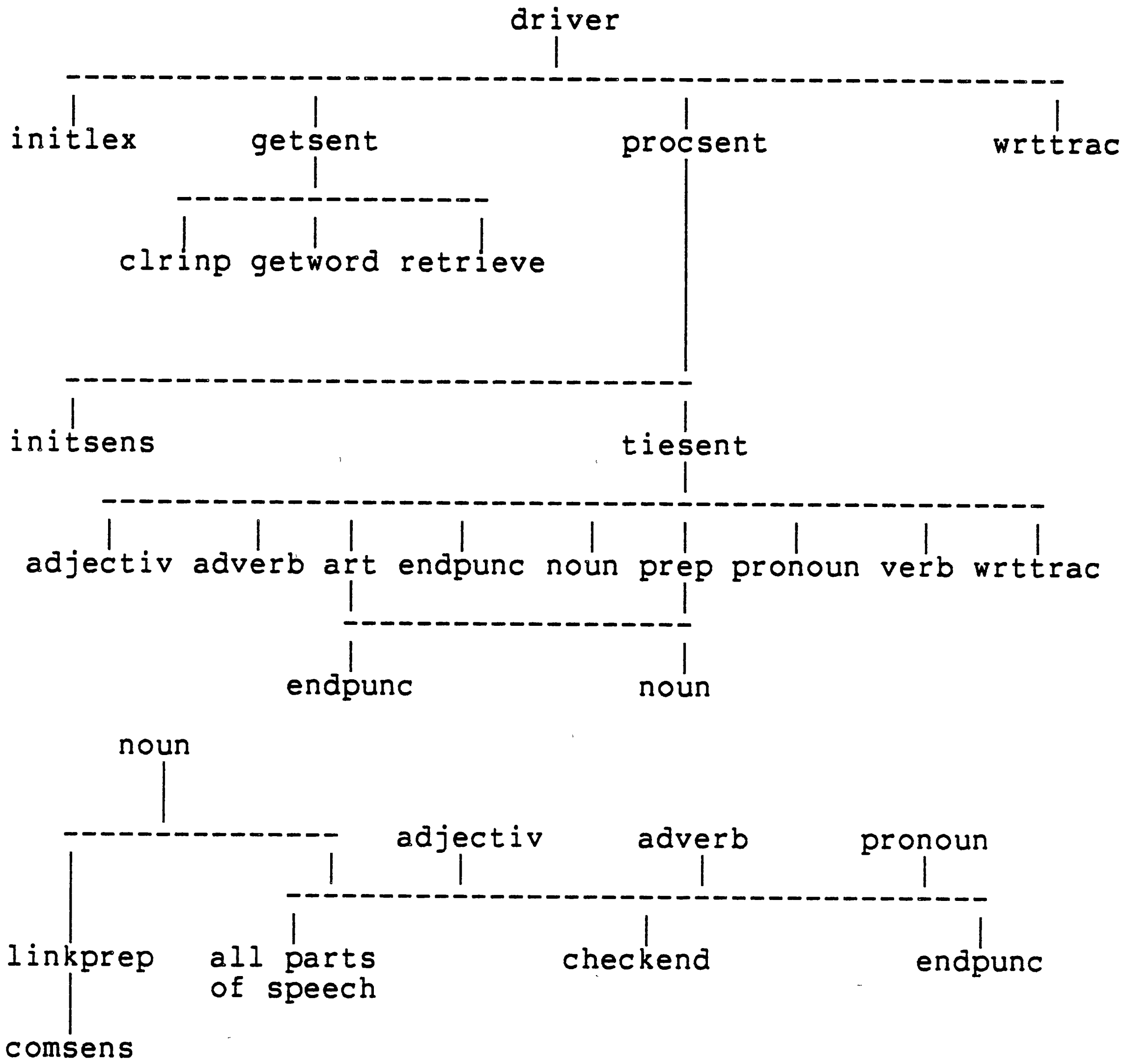
30. verb.c - Drives verb analysis

31. verblink.c - Links the subject, verb, and object

32. verbproc.c - Drives analysis of links among the words

33. wrttrac.c - Prints the sentence words and senses

MODULE LINKAGE DIAGRAM:



```
                            verb
                              |
    ----------------------------------------------------
     |              |                |                |
checkend         endpunc         verbproc         all parts
                                     |            of speech
                                     |            except
                                     |            verb
                                     |
   ----------------------------------------------------
   |                 |               |                |
asgnsens          linkart         linkint         verblink
                     |               |                |
                   matart          matart         linksbob
                     |               |                |
                     |               |            --------
                     |               |            |      |
                   tieart         tieart       matobj matsub
                                                 |      |
                                               tieobj tiesub
```

OPERATION: Cai is located in /grad/sam/thesis/german and may be invoked by the name "cai."

Cai is kept up-to-date by the use of a makefile located in the same directory.

ADDITIONAL INFORMATION:

Further documentation about the system may be found in the programs themselves, in the linklex system which maintains the lexicon located in /grad/sam/thesis/german/link, and in my thesis located in /grad/sam/thesis.

The ASCII source code of the programs may be obtained by sending a 9-track computer tape capable of 800 bpi or 1600 bpi to:

Oklahoma State University
Department of Computing and Information Sciences
Stillwater, Oklahoma 74078

VITA

Susan Ann Mengel

Candidate for the Degree of

Master of Science

Thesis: LEXICON DESIGN IN A GERMAN NATURAL LANGUAGE SYSTEM

Major Field: Computing and Information Science

Biographical:

Personal Data: Born in Indianapolis, Indiana, April 30, 1961, the daughter of Ralph H. and Jessie Mae Mengel.

Education: Graduated from Edmond Memorial High School, Edmond, Oklahoma, in May 1979; received Bachelor of Science degree in Computer Science from Central State University in May, 1982; completed requirements for the Master of Science degree at Oklahoma State University in December, 1984.

Professional Experience: Research Analyst, Department of Institutional Research, Central State University, May, 1979, to May, 1980; Computer Programmer, Department of Institutional Research, Central State University, May, 1980, to December, 1980; Teaching Assistant, Department of Computing and Information Sciences, Oklahoma State University, August, 1982, to July, 1984.

Professional Organizations: Association for Computing Machinery, Association for Computational Linguistics.