NATURAL LANGUAGE PROCESSING IN THE UNDERGRADUATE CURRICULUM

Mary Dee Harris Fosberg, Ph.D.
Department of Mathematical Sciences
Loyola University
New Orleans, LA 70118

Natural Language Processing is the manipulation by computers of languages used by humans for natural communication. It has been a part of the field of Computer Science since the 1950's. During the Red Scare of that era the possibility of having computers translate Russian documents of all sorts into English without intervention by humans seemed promising. However, attempts at machine translation revealed more about the infancy of computer processing of natural language and lack of understanding, even by linguists, of the nature of language in general, than about the secrets of the Communist threat. Despite the lack of success with machine translation projects at that time, researchers in several fields continued to investigate the problems of natural language processing: Linguists probing into the intricacies of understanding and using language, psychologists trying to determine how people think and learn and remember and how the use of language represents those activities, and computer scientists studying computer/user interfaces--how can humans store and retrieve information in computer memories in a form natural to themselves.

For many years predictions of the future development of the computer have included the scenario of the "person-on-the-street" being able to communicate with this amazing machine by merely typing a message--in English, of course--into a computer terminal, or--even better--by speaking a message to the control panel and being answered immediately in a pleasant, human voice from the computer. Any ordinary person, presumably with no extra-ordinary training, would be able to write programs for computers in his own natural language without any difficulty. Every home would have a computer or at least a terminal through which the family could communicate with the outside world acquiring information from libraries, retail stores, and news sources, to name only a few possibilities. And of course, the family robot could communicate in the family's language, just as C3P0 does in the movies.

 0-89791-036-2/81/0200/0196 $00.75

But to be realistic, as an educator I must ask: who is available to develop these marvelous systems? How many computer science students are being introduced to natural language processing to any extent comparable with the deluge of numerical methods and analysis and algorithms. Many computer science students are quite inexperienced with character manipulation and text processing, despite the fact that a large percentage of the information to be dealt with by computers is not numeric or even quantifiable. And natural language implies writing and verbal communication, both of which many computer people avoid at all costs.

Natural Language Processing is not an established area like data processing, for example. Payroll systems are many and varied, but the basic approaches to developing payroll software have been around for years. Natural Language Processing is still largely an area of research. But there have been some successful projects--restricted and limited, perhaps, but successful. Even attempts at machine translation have improved dramatically over the years. Certain techniques of manipulating language and methods of representing language information can be described and evaluated.

Why teach Natural Language Processing to undergraduates? Obviously graduate students could benefit from knowledge of an area primarily of research interest, but why undergraduates? First of all, graduate students start out being undergraduates, and as undergraduates become acquainted with the field of computer science as a whole. In studying the field they should be introduced to the major areas of knowledge and application of computer science and given practice in the skills necessary for solving the various types of problems encountered. I believe that natural language processing is an important area of computer science that should be introduced along with numerical algorithms and data base systems. Our society is producing natural language at an amazing rate, and the trend is toward more and more in the future. The traditional approach to handling text is to preprocess it: have a person identify keywords and abstract it so a computer can manipulate and "understand" it. But the largest quantity of Natural Language that is being produced has not been preprocessed, making that approach ineffective. Processing text as compared with natural language is easy; text is just another data structure with lots of variables: length, format, symbols. Natural language involves all the factors of text processing plus the problems of interpreting meanings.

COSC 440--Natural Language Processing

The course, Natural Language Processing, is an upper-level course for computer science majors. The prerequisites include Survey of Programming Languages and Data Structures, so most students have at least 21 hours of Computer Science. Because the course is only offered every two years, many students will have taken more than 21 hours. The course is organized into four primary sections: text processing, sentence generation, sentence analysis, and case studies.

Text Processing

The discussion of text processing was intended to be review; however I discovered that the students had little experience manipulating text. Various data structures for representing text were discussed; strings and arrays of characters were compared with respect to the algorithms required with each for the various primitive operations on text: insertion, deletion, concatenation, pattern matching, etc. Problems with data formatting were presented, such as variable field length and variable field inclusion. Several methods for formatting text were described including delimiters, codes, and bit maps. Several examples of complex machine-readable text were given such as WEBMARC, the MARC-formatted version of Webster's Seventh Collegiate Dictionary (Sherman, 1974). The project for this section of the course involved manipulation of unedited bibliographic references (on Natural Language Processing, of course). The students were required to design a format for the bibliographic material, then format the data, using the text editor on the computer system. When the data had been formatted, the students first produced a standard printout (prettyprinted), then either a KWIC index to the titles, an interactive keyword retrieval program or a selective sort on title, author or date of publication. The section on text processing was concluded by discussion of several important areas of application including lexicography, library science, content analysis, and automatic indexing. In concluding this section, the point was made that "text" need not be only written text by describing the Kurzweiler Reading machine for the blind and discussing techniques of voice production for telephone communications as examples of "verbal text".

Sentence Generation

The section of the course on sentence generation began with an introduction to linguistics grounded in the traditions of the schoolroom grammar with which students are already familiar. Some basic definitions from linguistics were presented: language, phoneme, morpheme, morphology, syntactics, and semantics. Noam Chomsky's theories of Transformational-Generative grammar provided the basis for the two programming projects on sentence generation. Based on Chomsky's earliest theories, the first project involved generating sentences from a simple phrase-structure grammar and a lexicon of words. The grammar is shown in Figure 1. The words in the lexicon were divided into the basic parts of speech: nouns, adjectives, adverbs, prepositions, articles, conjunctions, and verbs. The verbs were further subdivided into transitive, intransitive, copulative, and auxiliary verbs. The students' programs merely selected a word randomly from each of the appropriate sets of words to produce sentences without any attempt at making transformations for syntactic or semantic sense. As a result most of the sentences were nonsense, but many seemed quite poetic (see Figure 2). The words in the lexicon were taken mainly from the set of words used by Dylan Thomas during composition of "Poem on his Birthday," which had been the object of my research for several years. The random sentences retained a strong flavor of Thomas' poetry--which, I believe, reveals something about the nature of poetry in general.

The second sentence generation project using Chomsky's later theories about Transformational -generative grammars was more complex. The phrase-structure rules were replaced with the base component (See figure 3) composed of a set of categorial rules and a lexicon containing not only the vocabulary words in the language, but also information about the features of each word. These features provide syntactic, semantic, and phonological information about the words required for correct operation of the rules. The words were still divided into parts of speech as before, but contained additional coding to specify the following features:

1) Nouns were identified as animate or inanimate, human or non-human, concrete or abstract, and countable or non-countable. In addition, to simplify the phonological transformations, nouns which form irregular plurals (eg. woman, women) were tagged as such, and a set of the irregular plural forms created for table look-up.

2) The set of verbs included the three principal parts of speech for each: First person singular present indicative, first person singular past indicative, and past participle. In addition, an indication was included to specify whether the verb was transitive or intransitive. Coding the features for the verb required separate specifications for the type of subject the verb could take and the type of object it could take, if any. Thus the subject and the object were coded for animate or inanimate, human or non-human, and concrete or abstract.

3) Auxiliary verbs were separate from other verbs and included an indication of which principle part of the verb was required with each (eg. "did" + present indicative, "has" + past participle).

4) Articles were encoded singular or plural and definite or indefinite.

All other parts of speech--adjectives, adverbs, prepositions, etc.--remained the same. A set of names was added to allow substitution for singular human nouns.

For both of the sentence generation projects, the students were encouraged to design their programs to correspond to the linguistic theories being represented. Those students who had done that on the first sentence generation project had less trouble designing the program for the second project. They were expected to divide their programs into two primary sections for the second project; the first produced the deep structure by randomly selecting a verb, then randomly selecting nouns until an appropriate semantic match was found, following which other semantic features were determined (such as whether the subject was singular or plural and whether the verb was present tense or past tense). The second section of the program developed the surface structure of the sentence from the deep structure by transforming the words originally selected into proper syntactic form. (See figure 4 for examples.) The sentences produced were much more logical than the first project, but many semantic elements had still not been accounted for.

Sentence Analysis

The section of the course dealing with sentence analysis began with a discussion of parsing using the transformational-generative grammars as a paradigm. Several approaches to parsing were considered: Naomi Sager and Ralph Grishman's String Parser for Scientific Literature (Rustin, 1973) and Joyce Friedman's work with computer models of transformational grammars for instance. The discussion included general parsing techniques such as using reductions and action routines to find the patterns of such parts of speech. Parsing was presented as the opposite of sentence generation using rewrite rules of the phrase structure or categorial rules. Such techniques, which are highly successful in dealing with artificial languages--in particular, programming languages--do not work well with natural languages unless excessive restrictions are applied to the natural language. Thus, other approaches to sentence analysis had to be considered.

At this point in the class, case grammars were presented. Case grammars attack the problem of sentence analysis from the point of the verb being the focus of the sentence. A sentence at the deep structure level consists of a modality component and a proposition; the proposition is composed of a verb and all the cases related to that verb (See Figure 5). Using Robert F. Simmons' terminology, which is based on Celce-Murcia's work, the cases include one or two Causal-Actants (CA1 and CA2), a Theme, a Locus, a Source, and a Goal. The sentence,

"John broke the window with a hammer."

would be analyzed as:

```
((Modality:
            Tense, past;
            Mood, declarative;
            Essence, positive;
            Form, simple. . . ;)
  Proposition:  Break--CA1 John, Theme
the window,CA2 a hammer))
```

The program for sentence analysis required the student's program to accept sentences in English and analyze each one

in terms of case grammar, determining the modality of the sentence, the verb, and the various cases. (For the project the students used a different set of cases from Simmons and were not required to handle all possible modalities.) Again the students dealt with a set of rules and a lexicon. The rules (as shown in Figure 6) represent the deep structure possibilities, and the lexicon (now undivided, but in alphabetical order) contained for each word, identification of its possible syntactic functions (noun, verb, etc.), many of the same features from the previous project (animate, human, concrete, etc.), and in addition for each verb, a case frame. The case frame for a verb identifies which cases are allowed and/or required with a particular verb. All words in all forms are in the lexicon except for simple transformations such as plurals formed by adding "-s" to the noun; entries which are irregular or alternate forms point to the primary form of the word. Thus the entry "ran" would point to "run" from which all features would be derived. The lexicon is quite restricted in order to make the projects feasible within a part of the semester, but enough different features and classes of verbs are included to illustrate the important concepts involved in the theory of case grammars.

Case grammars can be represented by semantic networks with the words as the nodes and the semantic relations, the directed arcs. Thus a sentence analyzed with case grammar can be represented in an easily manipulated structure within a program. The students working on the sentence analysis project were encouraged to use semantic nets to represent sentences analyzed by case grammar. (LISP would have been an appropriate language for much of the course, but it was not available. Microlisp implemented on the Apple II was used to introduce the language, but that version of LISP is not robust enough for student projects.)

Case Studies

The last section of the course used case studies to illustrate applications of Natural Language Processing. Terry Winograd's SHRDLU was presented to illustrate a complete system; the program represents a robot that can manipulate blocks on a table top. SHRDLU accepts natural language input, interprets it in terms of what it knows, and responds both in natural language and through appropriate actions. Another illustrative system was W.A. Woods' Lunar Sciences Natural Language Information System, which provides the capability of natural language inquiries about a data base on lunar rock samples. Numerous other systems were discussed to show the levels of success attained with natural language processing systems.

To conclude the course, I discussed the implications of limiting ourselves to sentences as the natural language units to be manipulated; in other words, what is lost by ignoring larger units of language such as paragraphs or chapters in general text or lines and stanzas in poetry or scenes and acts in drama. Obviously some larger meaning is obscured, but considering the difficulties inherent just in dealing with the sentence as unit the limitation is justified in classroom exercises. We also examined the various problems associated with manipulating large volumes of information, such as disk access speed limitations vs. limitations of memory size. The course concluded with a rather philosophical evaluation of the projects, noting particularly the restrictions on English usage accepted in the course, as compared to the more general usage of "everyday" English language.

Implementation

The course was conducted primarily as a projects-oriented course with lectures held one or two times a week to acquaint students with the techniques required for the current project. I was available during the other one or two class periods per week for consultation with the students. Additional time would be required with a larger class. The grading was based on a contract agreed to at the beginning of the semester (See Figure 7). Six projects were required by students contracting for an "A" in the course (which all of my students chose): Two text processing programs, two sentence generation programs, one sentence analysis program, and a term project approved by the instructor. The term projects were primarily refinements to the sentence analysis project--adding new possibilities to allow for more general sentences. (Two students were interested in investigating voice generation and associated problems of verbal natural language.) The students were given a choice of programming languages to use for their projects, but were encouraged to choose a language with string functions included. They all chose either BASIC on the HP3000 or UCSD PASCAL on the Apple II. The data files for the various projects were all created on the HP3000 system and then down-loaded to the Apple II Pascal System for the students using the microcomputers. In addition to the programming projects assigned, students were required to read and report on some of the literature in the field, using the bibliographic references on Natural Language Processing as a starting point for their research. They also had to present the results of their term projects to the class.

The primary orientation of the entire course was on the algorithms and data structures needed to implement the projects. How can one represent meaning? Deep structures vs. surface structures? How are synonyms recognized in the data? What about homonyms? How does one decide between slow disk access and excessive memory usage? Are there any other options? All these questions were tied to problems of program design. Answering these--and many other--questions during the completion of their projects gave the students an opportunity to apply much of the knowledge gained from previous courses.

Conclusions

I have now taught Natural Language Processing as an undergraduate course for Computer Science majors two semesters at two different universities, and I have concluded that it is a valuable and enjoyable experience for the students. In addition to learning new programming techniques the students seem to benefit from learning more about their own natural language--English. Many observers have noted that computer people are frequently not verbally oriented, yet Computer Science educators are in agreement that our students need to be able to communicate with each other and with users, both orally and in writing. If a course such as Natural Language Processing can improve the students' awareness of their own language even a little, while at the same time providing information and sharpening skills relevant to computer science, the course must be considered worthy of inclusion in the curriculum. Students who have taken the course have enjoyed it very much; in fact the first time I taught the course the students often did not want to leave when the class was over. I cannot claim that this strange phenomenon occured due to my excellence in teaching because it has seldom happened in any other course of mine. But students seem to like talking about a subject on which they all feel like authorities--their own language.

Many of the areas discussed in the course relate to other Computer Science courses as well. Text processing is one important aspect of Data Structures in general, and accessing the data files in various ways for speed and efficiency relates to the File Processing Course. The study of programming languages has derived a great deal from linguistics; the terminology (syntax, semantics, grammar, parsing) was first used by the grammarians and linguists. BNF grammar notation is adapted from Noam Chomsky's phrase structure rules for describing the grammar of English. And many of the parsing methods used for artificial languages are similar to those used for natural languages. Thus in addition to providing the students with information about a generally ignored area of Computer Science, the course reinforces the knowledge acquired in other courses.

REFERENCES

Eugene Charniak & Yorick Wilks, Computational Semantics (New York, 1976).

Noam Chomsky, Aspects of the Theory of Syntax (Cambridge, Mass., 1965).

Noam Chomsky, Syntactic Structures (The Hague, 1957).

C. Fillmore, "The Case for Case," in Bach & Harms (eds.) Universals in Linguistics Theory (New York, 1968).

Marvin Minsky, Semantic Information Processing (Cambridge, Mass., 1968).

Randall Rustin (ed.), Natural Language Processing (New York, 1973).

R. R. Schank & K. M. Colby (eds.), Computer Models of Language and Thought (San Francisco, 1973).

Donald Sherman, "A New Computer Format for Webster's Seventh Collegiate Dictionary," Computers and the Humanities 8, 21-26.

R. F. Simmons, "Semantic Networks: Their Computation and Use for Understanding English Sentences," in Schank & Colby.

Donald E. Walker (ed.), Understanding Spoken Language (New York, 1978).

Terry Winograd, Understanding Natural Language (New York, 1972).

W. A. Woods, "Transition Network Grammars for Natural Language Analysis," Comm. ACM 13, 591-606.

GRAMMAR

Minimum:

$$S \longrightarrow NP + VP$$

$$NP \longrightarrow \left\{ \begin{array}{l} Art + Noun \\ Art + Adj + Noun \\ Adj + Noun \end{array} \right\}$$

$$VP \longrightarrow \left\{ \begin{array}{l} V \\ Aux + V \end{array} \right\} \quad (Adv)$$

$$V \longrightarrow \left\{ \begin{array}{l} Verbi \\ Verbt + NP \\ Verbc + Adj \end{array} \right\}$$

Extensions:

Prepositional phrases (adverbial or adjectival)
Compound sentences
Complex sentences
Compound nouns, verbs, adjectives, adverbs

Notation:

S: Sentence
NP: Noun Phrase
VP: Verb Phrase
Art: Article
Adj: Adjective
Adv: Adverb
Verbi: Verb (Intransitive)
Verbt: Verb (Transitive)
Verbc: Verb (Copulative)

Figure 1

SENTENCE GENERATION I

Examples of Computer Generated Sentences:

AN FURRED SNAKES SPURNS THE LINES.

THE FLESH SLAY THE BOMB.

AN WHIRLPOOL TOLLS A BELLED TUSK ALWAYS.

BROWN DYING WILL JOIN SAVAGE CHIMERS.

TUMBLING TRUTH CAST THE CLOUDS.

THESE MUSCLED GALAXIES CANNOT MELTS QUICKLY.

NEARING POINTS WILL FLY SLOWLY.

THEIR CANDLE-BLOWN TRADE WILL SCUD SMARTLY.

THE RAVISHED SKULL SEEM BRAINED QUICKLY.

THESE ENORMOUS TREETOP PLACES TANGLING LESS HATEFULLY.

THIS GOOD HILLS REMAINS GROSS.

THEIR SHATTERED WHALES WERE FOUNTAINOUS LOVELY.

PASSIONATE BANK MOURNS HARPTONGUED CHIMERS.

THIS FUMING TOWN AM SORROWING.

BODING PROPHETS DRAG FOUR FOXES ZIGZAG.

THESE POUNCING HOUSE IS EXULTING MANFULLY.

CONJURING BOULDERS HEAR LIVING PEACE EAST.

INCREDIBLE HERONS MUST OBEY BRAINED BLACKBERRIES.

Figure 2

NOAM CHOMSKY'S MODIFICATIONS OF HIS GRAMMAR

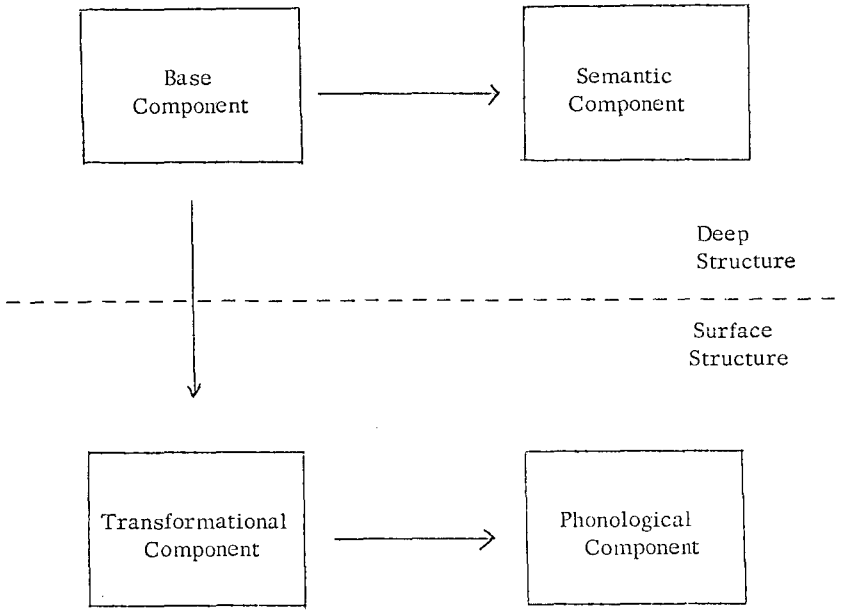

Figure 3

SENTENCE GENERATION II

Examples of Computer Generated Sentences:

TERESA DESPAIRED EARLY.

THE ALIVE THRONG SHRUNK THE TRAWLED TREETOP.

A CAVERNOUS SPINE BLOOMED INVINCIBLY.

SHALL THESE STRICKEN GEESE WADE?

THIS JAGGED SOLVER MUST BRAND THE ALOOF DANDELIONS.

MIGHT TUMBLEDOWN BEAST SIEZE THESE ANIMALS?

THE CROUCHED MOURNERS WOULD COUNT THE STARK SPEAR.

THESE JOYFUL DOGS DROWNED.

THESE WRECKED WRENS SLAY A GREY DAUGHTER.

THE DAUGHTER OUGHT TO HAVE SLID.

THE WOMEN KNOW PROPHETS.

LULLING GRASS BURSTS STARLIT RIPPLE.

THE GANG CAN SHOW THE NAKED CROWD.

A GOD SINGS.

HUNCHBACK WILL DRINK FORK.

THE PLUCKING SAKE PERCEIVES.

THE EMERALD MUST CELEBRATE.

THE TOSSED WOMEN STEAL THE SEALS.

Figure 4

CASE STRUCTURE OF A SENTENCE

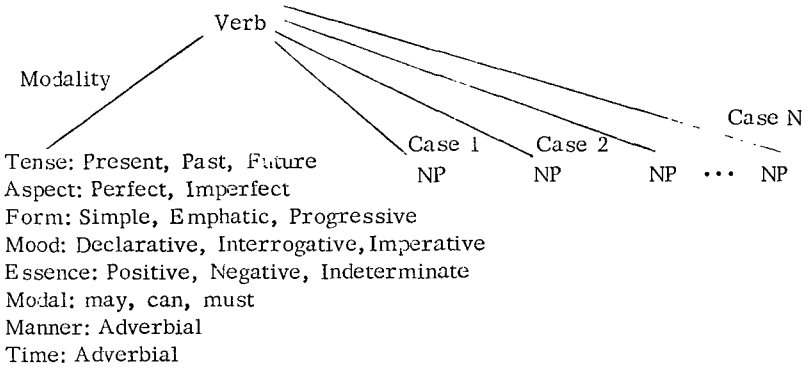

Figure 5

A PHRASE STRUCTURE GRAMMAR

S ⟶ Modality + Proposition
Modality ⟶ Tense, Aspect, Form, Mood, Modal, Manner, Time
Proposition ⟶ Vb + (CASEARGUMENT) *
Vb ⟶ run, walk, break, etc.
CASEARGUMENT ⟶ CASERELATION + NP/S
NP ⟶ [prep] + (DET) + (ADJ)* + (N) + N + (S/NP)
CASERELATION ⟶ CAUSALACTANT, THEME, LOCUS, SOURCE, GOAL.

Figure 6

GRADE CONTRACT

Computer Science 440

Natural Language Processing (NLP)

I. Programs
Text Processing I
Text Processing II
Sentence Generation I
Sentence Generation II
Sentence Analysis I
Term Project (to be approved--appropriate level for grade)

II. Reports
Written report on NLP system
Oral report on NLP system
Oral report on term project

III. Other
Additional library references for NLP bibliography
Format & entry of bibliographical references

I, the undersigned student, agree to complete adequately, and on time, the work described above for the grade of ____ in this course.

Any work turned in on time which is deemed inadequate for grade level contracted may be revised by student and regraded (one time only).

I, Mary Dee Harris Fosberg, Ph.D., agree to assign the grade of ____ to the undersigned student for adequate and prompt completion of the work described above.

Date _____

_____
Student

_____
Mary Dee Harris Fosberg, Ph.D.
Assistant Professor of
Computer Science

Figure 7