We have developed an approach to natural language processing in which the natural language processor is viewed as a knowledge-based system whose knowledge is about the meanings of the utterances of its language. The approach is oriented around the phrase rather than the word as the basic unit. We believe that this paradigm for language processing not only extends the capabilities of other natural language systems, but handles those tasks that previous systems could perform in a more systematic and extensible manner.

We have constructed a natural language analysis program called PHRAN (PHRasal ANALyzer) based in this approach. This model has a number of advantages over existing systems, including the ability to understand wider variety of language utterances, increased processing speed in some cases, structure from data structure, a knowledge base that could be shared by a language production mechanism, greater ease of extensibility, and the ability to store some useful forms of knowledge that cannot readily be added to other systems.

### 1.0 INTRODUCTION

The problem of constructing a natural language processing system may be viewed as a problem of constructing a knowledge-based system. From this orientation, the questions are as follows: What sort of knowledge does a system need about a language in order to understand the meaning of an utterance or to generate an utterance in that language? How can this knowledge about one's language best be represented, organized and utilized? Can these knowledge structures be achieved so that the resulting system is easy to add to and modify? Moreover, can the system be made to emulate a human language user?

Existing natural language processing systems vary considerably in the kinds of knowledge about language they possess, as well as in how this knowledge is represented, organized and utilized. Many of these systems are based on ideas about language that do not come to grips with the fact that a natural language processor needs a great deal of knowledge about the meaning of its language's utterances.

Part of the problem is that most current natural language systems assume that the meaning of a natural language utterance can be computed as a function of the constituents of the utterance. The basic constituents of utterances are assumed to be words, and all the knowledge the system has about the semantics of its language is stored at the word level (Fillmore, 1979) (Fleschke et al, 1979) (Wilks, 1975) (Woods, 1970). However, many natural language utterances have interpretations that cannot be found by examining their components. Idioms, collocations, and structural formulas are instances of large classes of language utterances whose interpretation require knowledge about the entire phrase independent of its individual words (Becker, 1975) (Mitchell, 1971).

We propose as an alternative a model of language use that comes from viewing language processing systems as knowledge-based systems that require the representation and organization of large amounts of knowledge about what the utterances of a language mean. This model has the following properties:

1. It has knowledge about the meaning of the words of the language, but in addition, the system's knowledge is about the meaning of larger forms of utterances.

2. This knowledge is stored in the form of pattern-concept pairs. A pattern is a phrasal construct of varying degrees of specificity. A concept is a notion that represents the meaning of the phrase. Together, this pair associates different forms of utterances with meanings.

3. The knowledge about language contained in the system is kept separate from the processing structures, that is, what we have knowledge to the understanding and production tasks.

4. The understanding component matches incoming utterances against known patterns, and then uses the concepts associated with the matched patterns to represent the utterance's meaning.

5. The production component expresses itself by looking for concepts that match the concept it wishes to express. The phrasal patterns associated with these concepts are used to generate the natural language utterance.

6. The database of pattern-concept pairs is shared by both the understanding mechanism and the mechanism of language production.

7. Other associations besides meanings may be kept along with a phrase. For example, a description of the contexts in which the phrase is an appropriate expression of a spoken utterance is stored.

PHRAN (PHRasal ANALyzer) is a natural language understanding system based on this view of language use. PHRAN reads English text and produces structures that represent its meaning. As it reads an utterance, PHRAN searches its knowledge base of pattern-concept pairs for patterns that best interpret the text. The concept portion of these pairs is then used to produce the meaning representation for the utterance.

PHRAN has a number of advantages over previous systems:

1. The system is able to handle phrasal language units that are awkwardly handled by previous systems but which exist in ordinary speech and common natural language texts.

2. It is simpler to add new information to the system because the representation and processing of patterns are kept separate. To extend the system, new pattern-concept pairs are simply added to the database.

3. The knowledge base used by PHRAN is declarative, and is in principle shareable by a system for language production (Such a mechanism is now under construction). Thus adding information to the base should extend the capabilities of both mechanisms.

4. Because associations other than meanings can be stored along with phrasal units, the identification of phrasal units may provide contextual clues not otherwise available to subsequent processing mechanisms.

5. The model seems to more adequately reflect the psychological reality of human language use.

### 2.0 PHRASAL LANGUAGE CONSTRUCTS

By the term "phrasal language constructs" we refer to those language units of which the language user has specific knowledge. We cannot present our entire classification of these constructs here. However, our phrasal constructs range greatly in flexibility. For example, fixed expressions like "by and large" or "the Big Apple" (meaning N.Y.C.), and lexical collocations such as "eye dropper" and "weak safety" allow little or no modification; idioms like "kick the bucket" and "bury the hatchet" allow the verb in them to appear in various forms: discontinuous dependencies like "look up" and "kick" are permitted even if they do not follow the production kick (e.g., "kick kicked, would have kicked") followed by some utterance denoting an object.

Our notion of a phrasal language construct is similar to a structural formula. Fillmore gives us knowledge to the understanding and production tasks.
be accommodated by the same phrasal pattern is essentially a conceptual one. Since each phrasal pattern in PHRAN is associated with a concept, if two meanings of phrases that appear different, they should be matched by different patterns. If the pattern for each of the phrases is similar enough that they seem to mean the same thing, then they should be accommodated by one pattern.

3.0 PHRAN

PHRAN (PHRasal ANalyzer) is an English language understanding system which integrates both generative and non-productive language abilities to provide a relatively flexible and extensible natural language understanding facility. While PHRAN does have knowledge about individual words, it is not limited to such knowledge nor is its processing capability constrained by a word-based bias.

Here are some examples of sentences PHRAN can understand:

- O'Flannery is encouraged by the amount of oil discovered in the Baltimore Canyon, an undersea trough 100 miles off the shore of New Jersey. (Newweek, Feb 1980)
- The young man was told to drive quickly over to Berkeley.
- If John gives Bill the big apple then Bill won't be hungry.
- Wills will drive Bill to The Big Apple if she is given twenty dollars.
- If Mary brings John we'll go to a Chinese restaurant.
- Will gets me a headache.

(The previous sentences are analysed by an uncompiled version of PHRAN on the DEC-20/4Q system at UC Berkeley in 2 to 3 seconds of CPU time).

At the center of PHRAN is a knowledge base of phrasal patterns. These include literal strings such as "so's off the shore of New Jersey. (Newsweek, Feb 1980)"

Pattern-concept pairs, a set of comprehension routines, and a routine which suggests appropriate pattern, concept pairs. PHRAN takes an input an English sentence, and as it reads it, PHRAN compares the sentence against patterns from the database. Whenever a pattern is found, the pattern is added to the list of patterns that PHRAN keeps and which will eventually contain the meaning of the sentence. Thus "CONCEPT" looks like:

```
{ <PERSON> - person, NF > } ...
```

Its initial condition is found to be satisfied by the first term in "CONCEPT" -- this fact is stored under that term and this fact will be added to the list of facts that this partial match continues. The term that was formed by reading "dropped" is now added to the list of concepts:

```
{ <PERSON> - person, NF >, <DROP - verb> }
```

PHRAN now checks to see if the pattern stored under the first term matches the term just added to "CONCEPT" too, and it does. This new fact is now stored under the last term.

Next the word "out" is read. The pattern suggestion mechanism is alerted by the occurrence of the verb 'drop' followed by the word 'out' and at this point it instructs PHRAN to consider the pattern:

```
{ <PERSON> - drop, NF >, <OUT - of> (school) }
```

The list in "CONCEPT" is checked against this pattern to see if it matches its first two terms, and since that is the case, this fact is stored under the second term. A term associated with "out" is now added to "CONCEPT":

```
{ <PERSON> - person, NF >, <DROP - verb> , <OUT - of> (school) }
```

The two patterns that have matched up to DROP are checked to see if the new term extends these. This is true only for the second pattern since the fact is stored under the next term. The pattern "(PERSON) - drop, NF, <DROP - object>" is discarded.

Now the word "of" is read. A term is formed and added to "CONCEPT". The pattern that matched up to OUT is extended by OF so the pattern is moved to the next term. The word "high" is read and a term is formed and added to "CONCEPT". Now this term under OF is compared against HIGH. It doesn't satisfy the next condition. PHRAN
reads "school" and the pattern suggestion routine presents PHRAN with two pattern concepts:

1. ["high" "school"]   [representation denoting a school for 10th through 12th grades,]  
2. [<adjective> <noun>]   [representation denoting a noun modified by adjective] 

Both patterns are satisfied by the previous term and this fact is stored under it. The new term is added to *CONCEPT*, now:

< [PERSON] (PERSON, NP) [DROP - verb] [OUT] [OF] [HIGH-SCHOOL] (school, NP) >

The pattern under OF is matched against the last term in *CONCEPT*. PHRAN finds a complete match, so all the matching terms are removed and replaced by the concept associated with this pattern.

*CONCEPT* now contains this concept as the final result:

< [(SCHOOLING (STUDENT JOHNI) (TERMINATION PREMATURE))] >

4.2 Pattern-Concept Pairs In More Detail

4.2.1 The Pattern

The pattern portion of a pattern-concept pair consists of a sequence of predicates. These may take one of several forms:

1. A word, which will match only a term representing this exact word.
2. A class name in parentheses; will match any term representing this class (e.g. "FOOD") or "PHYSICAL-OBJECT").
3. A pair, the first element of which is a property name and the second is a value; will match any term having the required value of the property (e.g. "Part-Of-Speech VERB").

In addition, we may negate a condition or specify that a conjunction or disjunction of several must hold.

The following is one of the patterns which may be suggested by the occurrence of the verb "give" in an utterance:

[(PERSON) (ROOT GIVE) (PERSON) (PHYSOB)]

4.2.1.1 Optional Parts

To indicate the presence of optional terms, a list of pattern concept-pairs is inserted into the pattern at the appropriate place. These pairs have as their first element a sub-pattern that will match the optional terms. The second part describes how the new term to be formed if the main pattern is found should be modified to reflect the existence of the optional sub-pattern.

The concept corresponding to the optional part of a pattern is treated in a form slightly different from the way we treat regular concept parts of pattern-concept pairs. As usual, it consists of pairs of expressions. The first of each pair will be places as is at the end of the properties of the term to be formed, and the second will be evaluated first and then placed on that list.

For example, another pattern suggested when 'give' is seen is the following:

[(PERSON) (ROOT GIVE) (PHYSOB) [(TO (PERSON) (TO (OPT-VAL 2 CD-FORM)))]

The terms of this pattern describe a person, the verb 'give', and then some object that 'give' describes the optional terms, consisting of the word to follow in a person description. Associated with them pattern is a concept part that specifies what to do with the optional part if it is there. Here it specifies that the second term in the pattern concept-pair is inserted into the pattern at the 70 slot in the conceptualization associated with the whole pattern.

This particular pattern need not be a separate pattern in PHRAN from the one that looks for the verb followed by the object transferred. We often have patterns with different optional parts to handle different verb forms, as is possible for expository purposes. Sometimes it is simpler to write the actual patterns separately, although we attach no theoretical significance to this dissection.

4.2.2 The Concept

When a pattern is matched, PHRAN removes the terms that match it from *CONCEPT* and replaces them with a new term, defined as described in section 4.1.4. This particular pattern need not be a separate pattern in PHRAN from the one that looks for the verb followed by the object transferred. We often have patterns with different optional parts to handle different verb forms, as is possible for expository purposes. Sometimes it is simpler to write the actual patterns separately, although we attach no theoretical significance to this dissection.

4.3 Pattern Manipulation In More Detail

4.3.1 Reading A Word

When a word is read PHRAN compares the patterns offered by the pattern suggesting routine with the list *CONCEPT* in the manner described in the example in section 4.1.3. It discards patterns that conflict with *CONCEPT* and retains the rest. Then PHRAN tries to determine which meaning of the word to choose, using the active patterns (those that have matched to the point where PHRAN has read). It checks if there is a particular meaning that will match the next slot in some pattern or, if no such definition exists, if there is a meaning that might be the beginning of a sequence of terms or meanings, as determined via a pattern-concept pair, will satisfy the next slot in one of the active patterns. If this is the case the word is chosen. Otherwise PHRAN defaults to the first of the meanings of the second slot in the pattern corresponding to the active pattern that matched to the current word. A new term is formed and if it satisfies the next condition in one of these patterns, the appropriate pattern moves to the pattern-list of the new term. If the next condition in the pattern indicates that the term specified is optional, then PHRAN checks for these optional terms, and if it detects that the concept is present, it checks to see if the new term satisfies the condition following the optional ones in the pattern.
4.3.2 A Pattern Is Matched

When a pattern has been matched completely, PHRAN constructs a list of all the other patterns on the pattern-list. When it has finished, PHRAN will take the longest pattern that was matched and will consider the possible meanings together. If there are several patterns of the same length that were matched, PHRAN will group all their meanings together.

New patterns are suggested and a disambiguation process follows, exactly as in the case of a new word being read.

For example, the words "the big apple", when recognized, will be two interchangeable meanings: "large fruit", the other being New York City. PHRAN will check the pattern of active words to determine if one of these two meanings satisfies the next condition in one of the patterns. If so, then that meaning will be chosen. Otherwise "a large fruit" will be the default, as it is the first in the list of possible meanings.

4.4 Adverbs And Adverbial Phrases

In certain cases there is need for slightly modified notions of pattern and meaning that we call special cases. For example, the word "quickly" may modify the meaning of the verb "eat" by changing the order of the words. In this case, PHRAN recognizes the adverb and modifies the pattern to reflect this change.

Thus PHRAN can handle constructs like:

quickly, John left the house.
slowly, John ate the apple.

Abbreviations are accepted in the input. Some patterns that are handled by specific patterns. For example, the negation of the verb want is handled as part of the pattern: "<person> (do) <want> not <verb> <object>". Thus PHRAN can handle the specific pattern: "<person> (do) not (want) <verb> <object>" which is associated with this interpretation.

4.5 Indexing And Pattern Suggestion

Retrieving the phrasal pattern matching a particular utterance from PHRAN's knowledge base is an important problem in the system. We have found that the use of specific patterns is helpful. PHRAN searches for patterns, of which there are two basic types: patterns consisting of words and phrases. The knowledge embodied is inaccessible to other mechanisms, in particular, production procedures.

Moreover, because Riesbeck's approach is word-oriented, it is difficult to incorporate phrasal patterns into his model. Some word of the phrase must have a routine associated with it that checks for that phrase. At best, this implementation is awkward.

An example of a natural language understanding system that produces declarative meaning representations is Riesbeck's "conceptual analyzer" (Riesbeck, 1974). Riesbeck's system (and the various systems that have followed) produce meaning structures as output. We contrast PHRAN with some of these.

5.0 COMPARISON TO OTHER SYSTEMS

There are a number of other natural language processing systems that either use some notion of pattern or incorporate phrase structures as output. We contrast PHRAN with some of them.

One of the earliest language understanding systems was a simulation of a paranoid mental patient that contains a natural language front-end (Parkinson et al., 1970). It receives a sentence as input and analyzes it in several separate stages. In effect, PARRY replaces the input with sentences of successively simpler form. In the simplified sentence PARRY searches for patterns, of which there are two basic types: patterns consisting of words and patterns consisting of phrases.

The knowledge embodied is inaccessible to other mechanisms, in particular, production procedures.

Moreover, because Riesbeck's approach is word-oriented, it is difficult to incorporate phrasal patterns into his model. Some word of the phrase must have a routine associated with it that checks for that phrase. At best, this implementation is awkward.

One of the earliest language understanding systems was a simulation of a paranoid mental patient that contains a natural language front-end (Parkinson et al., 1970). It receives a sentence as input and analyzes it in several separate stages. In effect, PARRY replaces the input with sentences of successively simpler form. In the simplified sentence PARRY searches for patterns, of which there are two basic types: patterns consisting of words and patterns consisting of phrases.

The knowledge embodied is inaccessible to other mechanisms, in particular, production procedures.

Moreover, because Riesbeck's approach is word-oriented, it is difficult to incorporate phrasal patterns into his model. Some word of the phrase must have a routine associated with it that checks for that phrase. At best, this implementation is awkward.
restricted to such an extent that many natural language processing problems need not be dealt with and other problems have solutions appropriate only to this context. In addition, SOPHIE does not produce any representation of the meaning of the input, and it makes more than one pass on the input ignoring unknown words, practices that have already been criticized.

The augmented finite state transition network (ATN) has been used by a number of researchers to aid in the analysis of natural language sentences (for example, see Woods 1970). However, most systems that use ATN incorporate one feature which we find objectionable on both theoretical and practical grounds. This is the separation of analysis into syntactic and semantic phases. The efficacy and psychological validity of the separation of syntactic and semantic processing has been argued at length elsewhere (see Schank 1975 for example). In addition, most ATN based systems (for example Woods' LUNAR program) do not produce representations, but rather, run queries of a data base.

In contrast to the systems just described, Wilks' English-French machine translator does not share several of their shortcomings (Wilks, 1973). It produces a representation of the meaning of an utterance, and it attempts to deal with unrestricted natural language. The main difference between Wilk's system and systems we describe is that Wilks' patterns are matched against concepts mentioned in a sentence. To recognize these concepts he attaches representations to words in a dictionary.

The problem is that this presupposes that there is a simple correspondence between the form of a concept and the form of a language utterance. However, it is the finding of this correspondence that is not simple. Most of the difficulties we are addressing in our work. In fact, since the correspondence of words to meaning is complex, it would appear that a program like Wilks' translator will eventually need the kind of knowledge embodied in PHRAN to complete its analysis.

One recent attempt at natural language analysis that radically departs from pattern-based approaches is Rieger and Small's system (Small, 1978). This system uses word experts rather than patterns as its basic mechanism. Their system acknowledges the enormity of the knowledge base required for language understanding, and provides a general way of addressing the relevant issues. However, the idea of producing a representation of the meaning of a sentence as a composite of individual words is about as far from our conception of language analysis as one can get, and we would argue, would exemplify all the problems we have described in word-based systems.

References

Becker, Joseph D. (1975). The phrase lexicon. In Theoretical Issues in Natural Language Processing, R. Schank and C.-O. Hoffman (Eds.), Cambridge, Mass.
Biranbaum, L. and Selfridge, M. (1970). Problems in conceptual analysis of natural language. Yale University Department of Computer Science Research Report 168.
Burton, Richard R. (1976). Semantic Grammar: An Engineering Technique for Constructing Natural Language Understanding Systems. IBM Research Report.
Fillmore, C.J. (1970). Innocence: A Second Idealization for Linguistics, in Proceedings of the Fifth Berkeley Language Symposium, Berkeley, California.
Hendrix, Gary G. (1977). The LIFMAN: A Guide to Building Practical Natural Language Interfaces. SRI International: AI Center Technical Note 158, Feb 1977.
Mitchell, T. F. (1971). Linguistic "Goings On": Collocations and Other Matters Arising on the Syntactic Record. Archivum Linguisticum 2 (new series) 76-99.
Perkinson, R.C., Colby, K.M., and Faught, W.S. (1977). Conversational Language Comprehension Using Integrated Pattern-Matching and Parsing. Artificial intelligence 9, 111-134.
Riesbeck, C. K. (1975). Conceptual analysis. In R. C. Schank Conceptual Information Processing. American Elsevier Publishing Company, Inc., New York.
Riesbeck, C. K. and Schank, R. C. (1975). Comprehension by computer: expectation-based analysis of sentences in context. Yale University Research Report 78.
Schank, R. C. (1975). Conceptual Information Processing. American Elsevier Publishing Company, Inc., New York.
Small, S. (1978). Conceptual language analysis for story comprehension. Technical Report No. 833, Dept. of Computer Science, University of Maryland, College Park, Maryland.
Wilks, Yorick (1973). An AI Approach to Machine Translation. In Computer Models of Thought and Language, R.C. Schank and K.M. Colby (eds.), W.H. Freeman and Co., San Francisco, 1975.
Woods, W. A. (1970). Transition Network Grammars for Natural Language Analysis. CACM 13, 591-606.
