PHRAN - A Knowledge-Based Natural Language Understander

Robert Wilensky and Yigal Arens

University of California at Berkeley

Abstract

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 a wider variety of language utterances, incressed processing speed in some cases, a clear separation of control 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 to ask are the following: What sort of knowledge does a system need about a language in order to understand the meaning of an utterance or to produce an utterance in that language? How can this knowledge about one's language best be represented, organized and utilized? Can these tasks 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. However, most 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 (Birnbaum et al, 1979) (Riesbeck et al, 1975) (Wilks, 1973) (Woods, 1970). However, many natural language utterances have interpretations that cannot be found by examining their components. Idioms, canned phrases, lexical 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, much of 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 notation that represents the meaning of the phrase. Together, this pair associates different forms of utterances with their meanings.
3. The knowledge about language contained in the system is kept separate from the processing strategies that apply this 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 in the data base 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 data-base 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 way to express its meaning may be stored. A person or situation strongly associated with the phrase may also be tied to it.

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 are found with great frequency in ordinary speech and common natural language texts.
2. It is simpler to add new information to the system because control and representation are kept separate. To extend the system, new pattern-concept pairs are simply added to the data-base.
3. The knowledge base used by PHRAN is declarative, and is in principle sharable 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 a phrase can 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", "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" permit varying positional relationships of their constituents. All these constructs are phrasal in that the language user must know the meaning of the construct as a whole in order to use it correctly.

In the most general case, a phrase may express the usage of a word sense. For example, to express one usage of the verb kick, the phrase "<person> <kick-form> <object>" is used. This denotes a person followed by some verb form involving 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, 1979). However, our criterion for determining whether a set of forms should

be accomodated by the same phrasal pattern is essentially a conceptual one. Since each phrasal pattern in PHRAN is associated with a concept, if the meanings of phrases are different, they should be matched by different patterns. If the surface structure of the phrases is similar and they seem to mean the same thing, then they should be accomodated 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:

* Oilmen are encouraged by the amount of oil discovered in the Baltimore Canyon, an undersea trough 100 miles off the shore of New Jersey. (Newsweek, 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.
* Wilma will drive Bill to The Big Apple if she is given twenty five dollars.
* If Mary brings John we'll go to a Chinese restaurant.
* Wilma gives me a headache.

(The previous sentences are analyzed by an uncompiled version of PHRAN on the DEC-20/40 system at UC Berkeley in from 2 to 9 seconds of CPU time).

At the center of PHRAN is a knowledge base of phrasal patterns. These include literal strings such as "so's your old man"; patterns such as "<nationality> restaurant", and very general phrases such as "<person> <give> <person> <object>".

Associated with each phrasal pattern is a conceptual template. A conceptual template is a piece of meaning representation with possible references to pieces of the associated phrasal pattern. For example, associated with the phrasal pattern "<nationality> restaurant" is the conceptual template denoting a restaurant that serves <nationality> type food; associated with the phrasal pattern "<person1> <give> <person2> <object>" is the conceptual template that denotes a transfer of possession by <person1> of <object> to <person2> from <person1>.

4.0 HOW PHRAN WORKS

4.1 Overall Algorithm

PHRAN is made up of three parts - a database of pattern-concept pairs, a set of comprehension routines, and a routine which suggests appropriate pattern-concept pairs. PHRAN takes as input an English sentence, and as it reads it from left to right, PHRAN compares the sentence against patterns from the database. Whenever a matching pattern is found, PHRAN interprets that part of the sentence that matched the pattern as describing the concept associated with the pattern in the pattern-concept pair.

4.1.1 Overview Of Processing -

When PHRAN analyzes a sentence, it reads the words one at a time, from left to right. It does just enough morphological analysis to recognize contractions and "'s"s. The pattern suggesting routine determines if any new patterns should be tried, and PHRAN checks all the new patterns to see if they agree with that part of the sentence already analyzed, discarding those that don't. A word's meaning is determined simply by its matching a pattern consisting of that literal word. Then a term is formed with the properties specified in the concept associated with the word, and this term is added to a list PHRAN maintains. PHRAN checks if the term it just added to the list completes or extends patterns that had already been partially matched by the previous terms. If a pattern is completely matched, the terms matching that pattern are removed and a new term, specified by the concept part of the pattern-concept pair, is formed and replaces the terms the pattern matched.

When PHRAN finishes processing one word it reads the next, iterating this procedure until it reaches the end of a sentence. At this point, it should end up with a single term on its list. This term contains the conceptualization representing the meaning of the whole sentence.

4.1.2 Overview Of PHRAN Patterns -

A pattern-concept pair consists of a specification of the phrasal unit, an associated concept, and some additional information about how the two are related. When PHRAN instantiates a concept, it creates an item called a term that includes the concept as well as some additional information.

A pattern is a sequence of conditions that must hold true for a sequence of terms. A pattern may specify optional terms too, the place where these may appear, and what effect (if any) their appearance will have on the properties of the term formed if the pattern is matched. For example, consider the following informal description of one of the patterns suggested by the mention of the verb 'to eat' in certain contexts.

```
{ pattern to recognize -
    [<first term: represents a person>
        <second term: is an active form of EAT>
            <OPTIONAL third term: represents food>]
 term to form -
    (INGEST (ACTOR <first term>)
            (OBJECT <third term, if present,
                                    else FOOD>)) }
```

Notice that the third term is marked as optional. If it is not present in the text, PHRAN will fill the OBJECT slot with a default representing generic food.

4.1.3 Simple Example -

The following is a highly simplified example of how PHRAN processes the sentence "John dropped out of school":

First the word "John" is read. "John" matches the pattern consisting of the literal "John", and the concept associated with this pattern causes a term to be formed that represents a noun phrase and a particular male person named John. No other patterns were suggested. This term is added on to *CONCEPT*, the list of terms PHRAN keeps and which will eventually contain the meaning of the sentence. Thus *CONCEPT* looks like

```
< [JOHN1 - person, NP] >
```

"Dropped" is read next. It matches the literal "dropped", and an appropriate term is formed. The pattern suggesting routine instructs PHRAN to consider the 'basic' pattern associated with the verb 'to drop', which is:

```
{ [<person> <DROP> <object>] [ ... ] }
```

Its initial condition is found to be satisfied by the first term in *CONCEPT* -- this fact is stored under that term so that succeeding ones will be checked to see if this partial match continues. The term that was formed after reading "dropped" is now added to the list. *CONCEPT* is now

```
< [JOHN1 - person, NP] , [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 occurence of the verb 'drop' followed by the word 'out', and at this point it instructs PHRAN to consider the pattern

```
{ [<person> <DROP> "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 secord term. A term associated with 'out' is now added to *CONCEPT*:

```
< [JOHN1 - person, NP] , [DROP - verb] , [OUT] >
```

The two patterns that have matched up to DROP are checked to see if the new term extends them. This is true only for the second pattern, and this fact is stored under the next term. The pattern [<person> <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 the pattern 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 patterns:

1. { ["high" "school"] [represention denoting a school for 10th through 12th graders] }

2. { [<adjective> <noun>] [representation denoting 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:

< [JOHN1 - person, NP] , [DROP - verb] , [OUT] ,
[OF] , [HIGH - adj] , [SCHOOL - school, noun] >

The two patterns are compared against the last term, and both are matched. The last two terms are removed from *CONCEPT*, and the patterns under OF are checked to determine which of the two possible meanings we have should be chosen. Patterns are suggested such that the more specific ones appear first, so that the more specific interpretation will be chosen if all patterns match equally well. Only if the second meaning (i.e. a school that is high) were explicitly specified by a previous pattern, would it have been chosen.

A term is formed and added to *CONCEPT*, which now contains

< [JOHN1 - person, NP] , [DROP - verb] , [OUT]
[OF] , [HIGH-SCHOOL1 - school, NP] >

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

CONCEPT now contains this concept as the final result:

< [($SCHOOLING (STUDENT JOHN1)
(SCHOOL HIGH-SCHOOL1)
(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 a member of 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 placed 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 physical object. The last term describes the optional terms, consisting of the word to followed by a person description. Associated with this 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 optional pattern should fill in the TO 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 recipient followed by the object transferred. We often show patterns without all the alternatives that are possible for expositional purposes. Sometimes it is simpler to write the actual patterns separately, although we attach no theoretical significance to this disposition.

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, as defined by the second part of the pattern-concept pair. For example, here is a pattern-concept pair that may be suggested when the verb 'eat' is encountered:

([(PERSON) (ROOT EAT) ([((FOOD))
(FOOD (OPT-VAL 1 CD-FORM))])]

[P-O-S 'SENTENCE
CD-FORM '(INGEST (ACTOR ?ACTOR) (OBJECT ?FOOD))
ACTOR (VALUE 1 CD-FORM)
FOOD 'FOOD])

The concept portion of this pair describes a term covering an entire sentence, and whose meaning is the action of INGESTing some food (Schank, 1975). The next two descriptors specify how to fill in variable parts of this action. The expression (VALUE n prop) specifies the 'prop' property of the n'th term in the matched sequence of the pattern (not including optional terms). OPT-VAL does the same thing with regards to a matched optional sub-pattern. Thus the concept description above specifies that the actor of the action is to be the term matching the first condition. The object eaten will be either the default concept food, or, if the optional sub-pattern was found, the term corresponding to this sub-pattern.

Sometimes a slot in the conceptualization can be filled by a term in a higher level pattern of which this one is an element. For example, when analyzing "John wanted to eat a cupcake" a slight modification of the previous pattern is used to find the meaning of "to eat a cupcake". Since no subject appears in this form, the higher level pattern specifies where it may find it. That is, a pattern associated with "want" looks like the following:

{ [<person> <WANT> <infinitive>]
[infinitive-subject (VALUE 1 CD-FORM)
.....] }

This specifies that the subject of the clause following want is the same as the subject of want.

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 up 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 whose meaning, as determined via a pattern-concept pair, will satisfy the next slot in one of the active patterns. If this is the case, that meaning of the word is chosen. Otherwise PHRAN defaults to the first of the meanings of the word.

A new term is formed and if it satisfies the next condition in one of these patterns, the appropriate pattern is moved 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 is convinced that they are not 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 continues checking 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 concept of its pattern-concept pair to be the meaning of the sequence. 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 have two possible meanings; one being a large fruit, the other being New York City. PHRAN will check the patterns active at that time 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 concept, the most prominent examples being adverbs and adverbial phrases. Such phrases are also recognized through the use of patterns. However, upon recognizing an adverb, PHRAN searches within the active patterns for an action that it can modify. When such an action is found the concept part of the pair associated with the adverb is used to modify the concept of the original action.

Adverbs such as "quickly" and "slowly" are currently defined and can be used to modify conceptualizations containing various actions. Thus PHRAN can handle constructs like:

John ate slowly.
Quickly, John left the house.
John left the house quickly.
John slowly ate the apple.
John wanted slowly to eat the apple.

Some special cases of negation are handled by specific patterns. For example, the negation of the verb want usually is interpreted as meaning "want not" - "Mary didn't want to go to school" means the same thing as "Mary wanted not to go to school". Thus PHRAN contains the specific pattern [<person> <do> "not" <want> <inf-phrase>] 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 that we have not yet solved to our complete satisfaction. We find some consolation in the fact that the problem of indexing a large data base is a neccesary and familiar problem for all knowledge based systems.

We have tried two pattern suggestion mechanisms with PHRAN:

1. Keying patterns off individual words or previously matched patterns.

2. Indexing patterns under ordered sequences of cues gotten from the sentence and phrasal patterns recognized in it.

The first indexing mechanism works but it requires that any pattern used to recognize a phrasal expressions be suggested by some word in it. This is unacceptable because it will cause the pattern to be suggested whenever the word it is triggered by is mentioned. The difficulties inherent in such an indexing scheme can be appreciated by considering which word in the phrase "by and large" should be used to trigger it. Any choice we make will cause the pattern to be suggested very often in contexts when it is not appropriate. In this form, PHRAN's processing roughly resembles ELI's (Riesbeck et al, 1975).

We therefore developed the second mechanism. The patterns-concept pairs of the database are indexed in a tree. As words are read, the pattern suggesting mechanism travels down this tree, choosing branches according to the meanings of the words. It suggests to PHRAN the patterns found at the nodes it has arrived at. The list of nodes is remembered, and when the next word is read the routine continues to branch from them, in addition to starting from the root. In practice, the number of nodes in the list is rather small.

For example, whenever a noun-phrase is followed by an active form of some verb, the suggesting routine instructs PHRAN to consider the simple declarative forms of the verb. When a noun-phrase is followed by the verb 'to be' followed by the perfective form of some verb, the routine instructs PHRAN to consider the passive uses of the last verb. The phrasal pattern that will recognize the expression "by and large" is found at the node reached only after seeing those three words consecutively. In this manner this pattern will be suggested only when neccessary.

The main problem with this scheme is that it does not lend itself well to allowing contextual cues to influence the choice of patterns PHRAN should try. This is one area where future research will be concentrated.

5.0 COMPARISON TO OTHER SYSTEMS

There are a number of other natural language processing systems that either use some notion of patterns or produce meaning structures as output. We contrast PHRAN with some of these.

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 descended from it) works by attaching routines to individual words. These routines are generally responsible for building pieces of a meaning representation. When a word is read by the system, the routines associated with that word are used to build up a meaning structure that eventually denotes the meaning of the entire utterance.

While our aims are much in the spirit of Riesbeck's analyzer, we believe there are both practical and theoretical difficulties inherent in his approach. For example, in Riesbeck's conceptual analyzer, specific understanding routines are needed for each word known to the system. Thus extending the system's vocabulary requires the creation and debugging of new code. In addition, these routines function only in the understanding process. The knowledge they embody is inaccessible to other mechanisms, in particular, to production procedures.

Moreover, because Riesbeck's approach is word-oriented, it is difficult to incorporate phrasal structures 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 to incorporate phrasal patterns is Colby's PARRY. PARRY is a simulation of a paranoid mental patient that contains a natural language front end (Parkinson et al, 1977). 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 used to interpret the whole sentence, and those used only to interpret parts of it (relative clauses, for example).

For PARRY, the purpose of the natural language analyzer is only to translate the input into a simplified form that a model of a paranoid person may use to determine an appropriate response. No attempt is made to model the analyzer itself after a human language user, as we are doing, nor are claims made to this effect. A system attempting to model human language analysis could not permit several unrelated passes, the use of a transition network grammar to interpret only certain sub-strings in the input, or a rule permitting it to simply ignore parts of the input.

This theoretical shortcoming of PARRY - having separate grammar rules for the complete sentence and for sub-parts of it - is shared by Hendrix's LIFER (Hendrix, 1977). LIFER is designed to enable a database to be queried using a subset of the English language. As is the case for PARRY, the natural language analysis done by LIFER is not meant to model humans. Rather, its function is to translate the input into instructions and produce a reply as efficiently as possible, and nothing resembling a representation of the meaning of the input is ever formed. Of course, the purpose of LIFER is not to be the front end of a system that understands coherent texts and which must therefore perform subsequent inference processes. While LIFER provides a workable solution to the natural language problem in a limited context, many general problems of language analysis are not addressed in that context.

SOPHIE (Burton, 1976) was designed to assist students in learning about simple electronic circuits. It can conduct a dialogue with the user in a restricted subset of the English language, and it uses knowledge about patterns of speech to interpret the input. SOPHIE accepts only certain questions and instructions concerning a few tasks. As is the case with LIFER, the language utterances acceptable to the system are

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's 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 system 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 fact that this correspondence is not simple that leads to the difficulties we are addressing in our work. In fact, since the correspondence of words to meanings 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 proposes a way of addressing the relevant issues. However, the idea of putting as much information as possible under 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 phrasal lexicon. In Theoretical Issues in Natural Language Processing. R. Schank and B.L. Nash-Webber (eds.). Cambridge, Mass.

Birnbaum, L. and Selfridge, M. (1979). 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. BBN Report No. 3453, Dec 1976.

Fillmore, C.J. (1979). Innocence: A Second Idealization for Linguistics. In Proceedings of the Fifth Berkeley Language Symposium, Berkeley, California.

Hendrix, Gary G. (1977). The Lifer Manual: A Guide to Building Practical Natural Language Interfaces. SRI International: AI Center Technical Note 138, Feb 1977.

Mitchell, T. F. (1971). Linguistic "Goings On"; Collocations and Other Matters Arising on the Syntactic Record. Archivum Linguisticum 2 (new series) 35-69.

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. 663, 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, 1973.

Woods, W. A. (1970). Transition Network Grammars for Natural Language Analysis. CACM 13, 591-606.