A Knowledge-based Approach to Language Processing: A Progress Report*

Robert Wilensky

Computer Science Division
Department of EECS
University of California, Berkeley
Berkeley, California 94720

Abstract

We present a model of natural language use meant to encompass the language-specific aspects of understanding and production. The model is motivated by the pervasiveness of non-generative language, by the desirability of a language analyzer ana a language production mechanism to share their knowledge, and the advantages of knowledge engineering features such as ease of extention and modification.

This model has been used as the basis for PHRAN, a language analyzer, and PHRED, a language production mechanism. We have implemented both these systems using a common knowledge base; we have produced versions of PHRAN that understand Spanish and Chinese with only changing the knowledge base and not modifying the program; and we have implemented PHRAN using the query language of a conventional relational data base system, and compared the performance of this system to a conventional LISP implementation.

1.0 INTRODUCTION

The need to cope with large quantities of knowledge has led to the emergence of "knowledge engineering issues in artificial intelligence. As it is desirable in practice for a system to be robust, modular, extensible, and easy to modify, a good deal of attention has been paid to the problem of designing systems that manifest these properties. Much of this research presumes that these goals will require ways of appropriately structuring the knowledge needed by the system, and thus is concerned with producing useful knowledge representations.

Constructing a natural language processing system may be viewed in exactly this manner. Both natural language analysis programs (those that input sentences and output meaning representations) and natural language production programs (those that look at meaning representations and output sentences) require a large body of knowledge, namely, knowledge about what the utterances of the language mean. However, the tendency has been to resist this point of view, and treat language knowledge as being somehow special. Thus while 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, the knowledge possessed by most of these systems has not been subjected to the sort of knowledge engineering analysis that knowledge of other kinds of knowledge-based systems have undergone.

We propose an alternative a model of language use that is derived from viewing language processing systems as knowledge-based systems. The knowledge that needs to be represented and organized here is the large amount of knowledge about what the utterances of a language mean. In this paper, I describe some of the theoretical underpinnings of the model, and then describe two programs, PHRAN and PHRED, that are based on these ideas. We have conducted a number of experiments with these systems that have some bearing on the utility of the model's presumptions, including testing these systems on other languages (Spanish and Chinese), and implementeing one of them in a relational data base system.

•Research sponsored by the Office of Naval Research under contract N00014-80-C-0732.

2.0 THE ASSUMPTIONS OF THE MODEL

2.1 The Importance Of Non-generative Language

Language user knows a great number of facts about what utterances of their language mean. That is, they know the meanings of a large number of words, and know the rules for relating these meanings to the occurrence of those words in an utterance. Moreover, they know the significance of a set of meaningful linguistic units that are not necessarily understood in terms of their components. We call all such units phrases . Included in this set are idioms, canned phrases, lexical collocations, cliches, structural formulas, and other non-generative language structures. For example, the language user needs to know the particular fact that "out of the blue" means unexpectedly, and that "<person1> bear <person2> a <sentiment> is a way of expressing a continued sentiment of one person toward another Tas in John bears Mary a grudge"), Our conjecture is that such units constitute a very considerable fraction of the language knowledge needed by an intelligent language processor.

In most theories of language, non-generative forms are usually considered to be theoretically uninteresting entities, or irritating special cases that violate the aesthetics of one's theory. However, if such structures do play a central role in language use, then most language processing is actually the application of these special case rules. This is precisely the point of view we take. That is, while our model allows for the more traditional, very general word-to-meaning mappings, these mappings play no priviledged role. Both generative and non-generative knowledge is represented and applied uniformly - the only difference is in the degree of abstractness of the knowledge encoded.

Once this view is taken, both language analysis and language production become kinds of data base management problems. The knowledge about the meanings of phrases of different shapes and abstractness is the data base. The problem is to represent this knowledge so it can be applied uniformly, and so it can be accessed correctly and efficiently for various language processing tasks.

While we believe that the notion of the primacy of phrase units is psychologically sound, we in fact take the data base notion quite seriously. That is, in terms of building practical, efficient language processing systems, the dominating problem may be one of data base management rather than computational complexity of the language processing algorithm. We will discuss the implications of this hypothesis below.

It should also be mentioned that we do not view our acceptance of a theory based on special cases as an abandonment of the hope of finding scientifically interesting generalisations to make about language. In fact, we believe there are principles *of* language use that can be derived from our approach. They are just not the principles one normally associates with language structure. Rather they are general principles of the application of this language knowledge. Interestingly, they are instances of more general principles that are also applicable to knowledge application that have no relation to language per se. The nature of some of these Princinles is discussed in Wilensky and Arena f 1(1980a)•

2.2 Sharable Knowledge Base

Language analysis and language production are of course very different problems. In language analysis, the task is to identify the meanings of incoming utterances; in production, the goal is to choose a language form the best encodes one s idea and intentions. However, in spite of these differring natures, it is reasonable to ask what knowledge these tasks share in common. As the language the user speaks and understands is more or less the same, it would not seem unreasonable that knowledge he uses to encode an idea in a sentence and knowledge he uses to understand the meaning of that very same sentence should somehow be related.

In our model, it is assumed that the knowledge uaed for analysis and for production is by and large the same. That is, there is only one data base of knowledge about the meaning of a language's forms. This knowledge base may be indexed and therefore accessed differently for different tasks, thus accounting for some of the asymmetries between the analysis and production. But the language knowledge used by both tasks is the same knowledge represented the same way.

'There are a number of reasons for believing that this assumption may be true for human language processors. Por example, people do not generally use words that they cannot understand, a possibility if their understanding and production knowledge were uncoupled. Also, it is certainly possible to talk about the meaning and use of a word or phrase independently or whether one is understanding or saying it. In fact, in our common understanding, separate analysis and production definitions for words are not recognized. That is, we do not normally believe that a word has one meaning when you say it and a separate meaning when you hear it.

However, the knowledge engineering reasons for this decision are more compelling. By having the knowledge of the two components be a shared data base, only one form of representation is needed. Moreover, the addition of new knowledge to this data base extends the capabilities of both systems simultaneously. One need only assert a piece of knowledge about the meaning of a phrase to the data base, and system will be able to understand that phrase when it occurs, as well as be able to use that phrase to express an idea for which the phrase is appropriate. As this requirement forces knowledge to be represented declaratively, the other benefits of such representations are enjoyed as well.

2.3 Benefits Of Declarative Representations

If language knowledge is to consist of one large data base used both for analysis and production, then it is imperative that this knowledge be stored in a highly declarative format. Only in this Banner can the same knowledge be used for two quite different tasks. Structuring the knowledge in this fashion entails several traditional knowledge engineering advantagea. For example, in this format, knowledge about the language is kept separate from the processing strategies that apply this knowledge to the understanding and production tasks. Thus adding new knowledge requires only adding new assertions to the data base, not writing and debugging new code.

In addition, other knowledge besides the meaning of a phrase can be easily associated with such declaractive representations. For example, the context in which a certain phrase is appropriate may be stored together with the meaning *of* that phrase; the analyzer can use such knowledge to help infer the context, the production mechanism to decide whether or not to use that phrase in a particular situation. Such additional information would be more difficult to introduce into a system that did not have phrasal knowledge stored as objects, i. e.# that wasn't phrasally oriented and didn't use declarative representations.

3.0 PHRAN AND PHRED

We have been developing this model of language use in two related programs, PHRAN (PHRasal ANalyter) and PHRED (PHRasal English Diction). PHRAN is a language understanding program written by Yigal Arens. It reads English sentences and produces representations from them that encode their meaning. PHRED is a natural language production mechanism developed by Steven Upstill. PHRED takes meaning representations as input and expresses them in English sentences.

Both PHRAN and PHRED share a common data base of language knowledge. This data base contains declarative representations about what the phrase of the English language mean. This knowledge is stored in the form of pattern-concept pairs. A pattern is a phrasal construction of varying degrees of specificity. For example, it may be an exact literal string, such as so s your old man"; it may be a pattern of limited flexibility such as "<nationality> restaurant , or <person> <kick> the bucket"; or it may be a very general phrase such as "<person> <give> <person> <object>".

The concept part of a pattern-concept pair is a conceptual template that represents the meaning of the associated phrase. The conceptual template is a piece of meaning representation with possible references to pieces of the associated phrasal pattern. The meaning representation we use is a variant of Conceptual Dependency (Schank, 1975). Together, these pairs associate different forms of utterances with their meanings. 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>.

PHRAN understands by reading the input text and trying to find the phrasal patterns that apply to it. AS it reads more of the text it may eliminate some possible patterns and suggest new ones. At some point it may recognize the completion of one or more patterns in the text. It may then have to chose among possible conflicting patterns. Finally, the conceptual template associated with the desired pattern is used to generate the structure denoting the meaning of the utterance. A detailed description of PHRAN is found in Wilensky and Arens (1980a,b; and in the article by Arens in these proceedings.

PHRED produces sentences that encode an idea by examining the same knowledge base. However, PHRED starts with a meaning representation it wishes to express, and tries to find conceptual templates that match it. If it finds more than one such template, it may have to chose between them. Then the phrasal pattern associated with the chosen conceptual template will be used to express the idea. Since these patterns may have variable pieces that relate to variable pieces of the conceptual template. PHRED must now find a way of expressing each subpart. The knowledge base of pattern-concept pairs is again consulted to find a way to do so: this knowledge is then used in a manner described by the initial pattern to form the appropriate mode of expression.

PHRAN and PHRED serve as the front and back end to various natural language processing systems. In general, PHRAN and PHRED perform that part of language processing that requires detailed knowledge of the specific language involved; the other components of the system perform reasoning based on more general, non-linguistic world knowledge. For example, PAM (Plan Applier Mechanism) is a story understanding program that can make inferences based on the goals and plans of the story characters. PAM also knows about the relevant saliency of the story components it encounters, so it can distinguish the points of a story from the story's less interesting parts.

PAM uses PHRAN to read the initial sentences of the story and produce representations of their meaning. After it has read the story, made the necessary inferences and recognized the story points, PAM uses PHRED to create a summary of that story in English by generating just those parts of the story representation that constitute the story points. The following example is meant to demonstrate some of PHRAN and PHRED's language processing capabilities:

Input to PHRAN:

JOHN GRADUATED COLLEGE. JOHN LOOKED FOR A JOB. THE XENON CORPORATION GAVE JOHN A JOB. JOHN WAS WELL LIKED BY THE XENON CORPORATION. JOHN WAS PROMOTED TO AN IMPORTANT POSITION BY THE XENON CORPORATION.

JOHN GOT INTO AN ARGUMENT WITH JOHN'S BOSS. JOHN'S BOSS GAVE JOHN'S JOB TO JOHN's ASSISTANT. JOHN COULDN'T FIND A JOB. JOHN COULDN'T MAKE A PAYMENT ON HIS CAR AND HAD TO GIVE UP HIS CAR. JOHN ALSO COULDN'T MAKE A PAYMENT ON HIS HOUSE, AND HAD TO SELL HIS HOUSE, AND MOVE TO A SMALL APARTMENT.

JOHN SAW A HIT AND RUN ACCIDENT. THE MAN WAS HURT. JOHN DIALED 911. THE MAN'S LIFE WAS SAVED. THE MAN WAS EXTREMELY WEALTHY, AND REWARDED JOHN WITH A MILLION DOLLARS. JOHN WAS OVERJOYED. JOHN BOUGHT A HUGE MANSION AND AN EXPENSIVE CAR, AND LIVED HAPPLY EVER AFTER.

After processsing by PAM...

Summary generated by PHRED:

JOHN WORKED FOR THE XENON CORPORATION.
THE XENON CORPORATION FIRED JOHN.
JOHN COULD NOT PAY FOR HIS HOUSE AND HIS CAR.
JOHN WAS BROKE.
A MAN GAVE JOHN SOME MONEY.
M W Fife CAR AND A NEW HOUSE.

In addition, PHRAN has been able to understander utterances of considerably greater complexity. For example, the following are examples of sentences PHRAN can understand, taken from Newsweek (March, 1980):

> Oilmen are encouraged by the amount of natural gas discovered in the Baltimore Canyon, an undersea trough about 100 miles off the New Jersey coast.
>
> Tenneco, one of "59 companies engaged in drilling in the area, thinks it's leased tract contains a marketable supply of gas.

As the knowledge base shared by PHRAN and PHRED contains phrasal patterns of various levels of abstraction, each mechanism handles generative as well as non-generative constructs with a single processing strategy. Again, since their knowledge base is in fact shared, adding a single item to both data bases makes that item usable both to the language understanding and language production components.

Both PHRAN and PHRAN are written in UCILISP and run on a KL-10 at Berkeley. Their knowledge base contains over three hundred patterns. PHRAN averages about 1 cpu-sec to analyse a sentence; PHRED about the same to generate one from an underlying concept. No attempt has been made to optimize these programs, which are both currently uncompiled code.

3.1 PHRAN And PHRED In A Model Of Language

PHRAN and PHRED represent that part of our language use apparatus that is language specific. As such, they represent separable, but not autonomous, components of the entire language processing facility. This is a rather different breakdown than the more traditional view of language that separates it into structural, meaning, and use components; in our model, syntactic, semantic and pragmatic knowledge may all be intertwined both functionally and structurally within a single pattern.

Other researchers that eschew these distinctions, for example, Schank, Lebowitz, and Birnbaum (1980), emphasize the importance of integrated functions to the point where they no longer recognize the existence of a language specific component. In contrast, we wish to preserve a level of language processing that is distinct from other sorts of knowledge application. Namely, this is the level of application of language specific knowledge as embodied by the PHRAN-PHRED knowledge-base. Our work suggests to us that the processing involved here is quite different from that required to apply other forms of world knowledge. For example, the kind of processing needed to understand the relation between sentences of a text appears to be unrelated to that needed to understand the way in which words of a sentence join together to produce a meaning.

The importance of this distinction is two-fold. First, it indicates that while language understanding is highly integrated with non-linguistic processing, that a separable level of language-specific processing is still isolatable. Secondly, since these other processes are viewed as being different in nature from PHRAN and PHRED, then we cannot view these programs as models for the entire process. That is, an integrated understander will contain components that are not designed along the same lines as PHRAN; a general production mechanism will contain parts that do not resemble the control structure of PARED.

4.0 SOME EXPERIENCE WITH THE MODEL

Although it may be too early to make a definitive evaluation of the model, our preliminary, qualitative results are encouraging. For example, the addition of new knowledge to the system is how rather straightforward. The sentences quoted above from Newsweek contain some vocabulary and phrases with which PHRAN was not familiar at the time we intended to process them. Getting PHRAN to work on them was accomplished merely by asserting the missing knowledge into the data base and indexing it (by hand). No new code had to be written or old code changed. Thus making these additions took a relatively short period of time (a few hours).

In this section I describe some experiences we have had in developing PHRAN and PHRED, as well as some experiments we *nave* made in using the model for other languages, and an alternative implementation of PHRAN using a relational data base system.

4.1 Sharing PHRAN*s Knowledge Base

Work on PHRAN began about a year before PHRgp. Thus PHRED was essentially designed using PHRAN's knowledge base. While their were some biases in this data base because it was somewhat more PHRAN-oriented, these proved to be technical rather

than theoretical problems. For example, the association between a variable part of a pattern and the part of the concept with which it is associated was kept in the form or a list of correspondences. However, in production, PHRED often needs to know that a particular pattern should be used only when two slots of a concept are filled with the same conceptualisation. For example, the pattern "<person> take <object>" is only applicable wnen the actor and recipient of the concept part are the same.

The problem here is that this equivalence is only implicit in the information PHRAN needs to fill slots during understanding. That is, PHRAN only needs to Know that the subject goes into the actor slot and the subject goes into the recipient slot, but it need not realise that these slots will therefore contain the same filler. However, this is exactly the information that PHRED needs to know it should use this pattern. This is a problem because PHRED must make a deduction here while PHRAN does not, or alternatively, we could add this derivable fact. However, this starts violating the idea that there would be no specialized knowledge for understanding and production.

The problem is not serious in any case. Moreover, it could easily be fixed by using pattern matching-type variables to encode the associations: this would establish a correspondence that is not biased more for one program or tne other. We have not bothered to do so only because it was easier to mildly violate one of our principles than to re-write all the patterns. Thus far, all the problems we have encountered with the representation of the data base have been of this sort. While this is encouraging we probably have not been working on PHRED long enough yet to be able to state categorically that no more significant problems will arise.

4.2 Spanish PHRAN

In addition to being easily extended, knowledge-engineered systems should be easily modifiable. One way in which someone may want to modify PHRAN is to make it work on another language. To begin with, we had no a priori feeling about now much of PHRAN is dependent upon aspects of the English language and how much could be generalized to other languages. Also, it seems reasonable that for some languages quite different from English, the pattern-based nature of PHRAN might be less well-suited, and for some, perhaps better suited.

Thus while we have were not in a position to make claims about linguistic universals, it did seem reasonable to expect that PHRAN should be easily adaptable to languages fairly close to English. Moreover, we are interested in now much the basic structure of PHRAN should carry over to languages that aren't quite so similar. To explore out these questions, Mike Morgan, a graduate student, has attempt to produce both a Spanish and a Chinese version of PHRAN simply by changing the pattern-concept data base. Major modification of the program itself was not considered fair play.

The Spanish version proved to be successful in a number of ways. First, we found that it was possible to rewrite most of the patterns into phrases of another language without having the knowledge encoder learn anything about the inner workings of the program. Thus roost of PHRAN s knowledge was converted into Spanish language knowledge in a few weeks of this sort of coding. This is particular encouraging since no effort was made to make PHRAN accessible to the naive user. We feel that this suggests that a system like PHRAN could be designed to allow fairly easy construction of a language processor for a new language, or to allow for the addition of special purpose phrases or jargon by some user who was not an expert AI programmer.

Thus far, almost all of PHRAN's knowledge base has been converted into Spanish. Actually, the Spanish version contains many patterns that do not occur in English and omits those that do not make sense in Spanish. Thus there is not a one-to-one correspondence between the data bases, but the amount of knowledge they contain is roughly the same. While we have encountered some problems in making this conversion, so far. these have all been technical problems with PHRAN per se. That is, these are known deficiencies with the basic mechanism of PHRAN that are easily correctable, but which have not yet been fixed. Although these manifest themselves more in the Spanish version, they do not seem to represent problems that are particular to Spanish processing. Probably the reason that they nave arisen more prominently in the Spanish version is due to the frequency of the kinds of constructions involved.

For example, the direct and indirect pronouns in Spanish can come in three types: 1) indirect-pronoun verb. *2)* direct-pronoun verb, and 3) indirect-pronoun direct-pronoun verb. Since some indirect pronouns and direct pronouns are the same word, care must be taken to differentiate between cases 1 and 2. This is done by specifying case-like patterns associated with each verb that require some particular kind of object.

The following are some examples of Spanish sentences this version of PHRAN can process. These examples were based on sentences in an elementary Spanish language textbook. They are chosen primarily to demonstrate those aspects of Spanish processing that do not arise in English PHRAN, such as the problems involving pronouns mentioned above. The actual Spanish input is provided, along with a literal translation. Tne representation produced by PHRAN is then shown, followed by some timing data (the output has been edited for understandability):

```
Input:  EL  RESTAURANTE CHINO   QUE  ESTA EN
       (The restaurant  Chinese that is  in

        BERKELEY HA  ESTADO VENDIENDO
-       Berkeley has been   selling

        HAMBURGUESAS RAPIDAMENTE POR DIEZ PESETAS.
-       hamburgers   rapidly     for 10   pesetas)
Output:
((RESTAURANT (OBJECT RESTAURANTE1))
 (CHINESE (OBJECT RESTAURANTE1))
 (GROUP (OBJECT HAMBURGUESAS1) (MEMBER HAMBURGER))
 (MONEY (OBJECT MONEY1) (AMOUNT 10))
 (IS (ACTOR RESTAURANTE1)
     (PLOC (PROX (LOCATION BERKELEY))))
(ATRANS (OBJECT MONEY1) (TO RESTAURANTE1))
(ATRANS (TENSE PROGRESSIVE-OCCURRENCE-PRESENT)
        (ACTOR RESTAURANTE1)
        (OBJECT HAMBURGUESAS1)
        (FROM RESTAURANTE1)
        (MODE FAST))
3729 msec CPU, 4650 msec clock, 11222 conses

Input:  LA  HIJA     DE EL  CAMARERO QUIERE TU
       (The daughter of the waiter   wants  your

        MANZANA
-       apple)

Output:
((APPLE (OBJECT MANZANA1))
 (FEMALE (OBJECT HIJA2))
 (PERSON (OBJECT HIJA2))
 (PERSON (OBJECT CAMARERO1))
 (WAITER (OBJECT CAMARERO1))
(CHILD-OF (ACTOR CAMARERO1) (OBJECT HIJA2))
(POSS (ACTOR *THOU*) (OBJECT MANZANA1))
(GOAL (TENSE PRESENT)
      (PLANNER HIJA2)
      (OBJECTIVE (POSS (ACTOR HIJA2)
                       (OBJECT MANZANA1))))
2771 msec CPU, 3400 msec clock, 7184 conses
```

```
Input:  YO VOY A  TENER QUE  IR A  CHICAGO
       (I  go  to have  that go to Chicago
        i. e., I'm going to have to go to Chicago)

Output:
(OBLIGATION
 (TENSE FUTURE)
 (ACTOR *EGO*)
 (OBLIGAND (PTRANS (TENSE FUTURE) (ACTOR *EGO*)
                   (OBJECT *EGO*) (FROM *HERE*)
                   (TO (PROX (LOCATION CHICAGO))))))
3457 msec CPU, 3950 msec clock, 4921 conses

Input:  LA  PERSONA QUE  ESTA EN CHICAGO QUIERE
       (The person  that is   in Chicago wants

        IR A  BERKELEY
        go to Berkeley)

Output:
((PERSON (OBJECT PERSONA1)))

(GOAL (TENSE PRESENT)
      (PLANNER PERSONA1)
      (OBJECTIVE
       (PTRANS (TENSE FUTURE) (ACTOR PERSONA1)
               (OBJECT PERSONA1) (FROM *HERE*)
               (TO (PROX (LOCATION BERKELEY))))))

(IS (ACTOR PERSONA1)
    (PLOC (PROX (LOCATION CHICAGO))))
2999 msec CPU, 3716 msec clock, 6470 conses

Input:  LAS MANZANAS ROJAS QUE  ESTABAN EN LA  CAJA
       (The apples   red   that are     in the box

        FUERON VENDIDAS POR JUAN
        were   sold     by  John)

Output:
((MALE (OBJECT JUAN1)) (PERSON (OBJECT JUAN1))
 (PHYSOB (OBJECT CAJA1)) (BOX (OBJECT CAJA1))
 (UNSPECIFIED (OBJECT IT3)) (GROUP (OBJECT IT3))
 (MONEY (OBJECT MONEY1))
 (GROUP (OBJECT MANZANAS1) (MEMBER APPLE))
 (COLOR (OBJECT MANZANAS1) (HUE RED)))

(ATRANS (ACTOR IT3) (OBJECT MONEY1)
        (TO JUAN1) (FROM IT3))
(ATRANS (ACTOR JUAN1) (OBJECT MANZANAS1)
        (TO IT3) (FROM JUAN1))
(IS (ACTOR MANZANAS1)
    (PLOC (INSIDE-OF (OBJECT CAJA1))))
```

4.3 Chinese PHRAN

Chinese is an interesting test of PHRAN because the language consists of a relatively small number of words, and a great deal of phrasal productivity. A Chinese version of PHRAN is being constructed, again by only changing the pattern-concept pair data base. The Pin-Yin romanization is used (i. e., Beijing as opposed to Peking), to, denote Chinese words, with suffixed numbers (1-4) denoting tones. Currently. Chinese PHRAN has all 1596 Chinese words in it, although it only knows the significance of a small percentage of the meaningful patterns.

For example, concepts often lexicalized in English can be denoted in Chinese by putting words together. Examples included so far in Chinese PHRAN are "xue2 sheng2" (student) and "xian1 sheng1" (teacher). Other more complex patterns also abound. For example, the pattern "<PLACE> ren2" means "person from that place"; "<PLACE> hua4" -> "the language spoken in that place"; "<COUNTRY-HEAD> guo2 -> "that country"; "<COUNTRY-HEAD> wen2" -> "language spoken in country". Thus "zhong1 guo2 ren2 means a person from China", and both "zhong1 guo2 hua4" and "zhong1 wen2" mean "Chinese".

Other examples include "Chu2 le ... (yi3 wai4) ...". Here (yi3 wai4) indicates an option which gives no added meaning. It corresponds to "Except for ..., ...". "Yin1 wei1 ..., suo3 yi3 ..." corresponds to "Because ..., so ...."

The following are examples of the Chinese sentences PHRAN can handle:

```
Input:  NI3   PENG2 YOU3 DE1 PENG2 YOU3 YOU3 YI2 GE4
       (Your  friend's     friend     has  a
        FA3 GUO2 FAN4 GUAN3 MEI2 YOU3
        France   restaurant not  have
        i. e., does he have one)
Output:
((FRIEND (ACTOR *YOU*) (OBJECT PERSON3))
 (FRIEND (ACTOR PERSON3) (OBJECT PERSON4))
 (SPECIFICATION (OBJECT PERSON4) (SPEC SPEC2))
 (RESTAURANT (OBJECT RESTAURANT1) (TYPE FRENCH)))

(POSSESS (MOOD INTERROGATIVE)
         (ACTOR PERSON4) (OBJECT RESTAURANT1))
2860 msec CPU, 3650 msec clock, 5960 conses

Input:  XIAN1 SHENG1 SHI4 ZHONG1 GUO2 REN2
       ( Teacher    is     China    person)
Output:
((PERSON TEACHER3)
 (TEACHER TEACHER3)
 (PERSON PERSON3)
 (ORIGIN (OBJECT PERSON3) (LOCATION CHINA)))
(IS (SUBJECT TEACHER3) (COMPLEMENT PERSON3))
775 msec CPU, 1084 msec clock, 1429 conses

Input:  NEI4 BEN3 SHU1 HEN3 YOU3 YI4 SI1
       (That      book very interesting)
Output:
((BOOK (OBJECT BOOK1)))

(IS (OBJECT BOOK1) (STATE-NAME QUALITY)
    (VALUE INTERESTING) (DEGREE VERY))
1028 msec CPU, 1250 msec clock, 2068 conses
```

We expect that it would be more difficult to extend PHRAN to languages that have a great deal of morphological structure. In these cases, our idea of a pattern will probably have to be extended to allow for the description of components of individual words. It is not clear at this point whether this will present any serious problems for the basic PHRAN processing structure.

5.0 AI AND RELATIONAL DATA BASES

By representing the knowledge about the meaning of a language's utterances declaratively, we claim to have reduced at least part of the natural language processing problem to a problem or data base manipulation. If so, then rather than implementing one's own data base in LISP, conventional data base management systems may be useful when the knowledge to be organized becomes large enough.

We decided to test this idea by actually implementing a version of PHRAN in a conventional data base system. Fred Mueller, a graduate student, essentially re-wrote PHRAN in EQUEL. a query language for the INGRES data base system (Held, Stonebraker, and Wong, 1975) developed at Berkeley. INGRES is based on the relational model, and runs on a VAX 11/780. To get reliable measurements. PHRAN was implemented in FRANZ LISP on this system as well. Test were run to compare the relative performance of the systems on various size data bases.

While the results are somewhat subject to varying interpretations, they can be summarized as follows: First, it was possible in fact to re-write PHRAN in INGRES, although there are some minor difficulties in encoding LISP-based data structures

into a language that does not support pointers. Second, the LISP version is considerably faster when the data base of pattern-concept pairs is small (about 11 times faster). However, when the data base is large (2000 words and 500 patterns), the EQUEL version is about 3 times faster than the LISP version.

These figures are not too surprising in that data bases systems are designed to handle large disk files and therefore should perform better than arbitrary LISP programs accessing a disk. However, these results are equivocal because (1) loading the data base was charged to some of the LISP execution times. (2) little attempt has been made to optimise the LISP version, and (3) the performance of INGRES is actually worse than reported, as the figures presume about a factor of four speed-up that should result from very simple changes to INGRES that are in the making.

The details of these numerical results are not so much the point, however. The speed-up figures are probably conservative in any case. The real point is that a great deal of attention to this sort of performance issue will be paid to general data base systems. If production versions of AI programs can be transferred to these systems to take advantage of this work, then a great deal of duplication of effort may be avoided. By engineering PHRAN in the manner that we have, we can take advantage of such technology as it becomes available without having to entirely re-design our system for a different implementation.

6.0 SUMMARY

We have presented a model of natural language use meant to describe the language-specific aspects of language understanding and production. The model is motivated by the pervasiveness of non-generative language, by the desirabity of a language analyzer and a language production mechanism to share their knowledge, and by the need for knowledge engineering advantages sucn as ease of extention and modification.

The basis of the model is a declarative knowledge base of pattern-concept pairs. This knowledge base is shared by PHRAN, a language analyzer, and PHRED, a language production mechanism. PHRAN matches patterns against sentences and uses the associated concepts to represent their meaning; PHRED matches representations for ideas against the concept parts of these patterns, and combines the pattern part to express their meaning.

We have experimented with these systems, particularly with PHRAN, by trying to convert it to other language, such as Spanish ana Chinese, and by implementing PHRAN in a relational data base system. The success we have had so far with these enterprises indicates to us that knowledge engineering principles are useful in the design of flexible, efficient, and theoretically interesting natural language processing systems.

References

1] Becker, Joseph D. (1975). The phrasal lexicon. In Theoretical Issues in Natural Language Processing. R. Schank and B.L. Nash-Webber (eds.). Cambridge, Mass.

2] Goldman, Neil (1975). Conceptual generation. In R. C. Schank, Conceptual Information Processing. American Elsevier Publishing Company, Inc., New York.

3] Held, G. D., Stonebraker, M. R., and Wong, E. (1975). INGRES - A relational data base system. AFIPS Conference Proceedings vol. 44, NCC.

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

5] Kay, Martin (1975). Syntactic Processing and Functional Sentence Perspective. In Theoretical Issues in Natural Language Processing, R. C. Schank and B. Nash-Webber (eds.). Cambridge, Mass.

6] Riesbeck, C. K. (1975). Conceptual analysis. In R. C. Schank, Conceptual Information Processing. American Elsevier Publishing Company, Inc., New York.

7] Riesbeck, C. K. and Schank, R. C. (1975). Comprehension by computer: expectation-based analysis of sentences in context. Yale University Research Report 78.

8] Schank, R. C. (1975). Conceptual Information Processing. American Elsevier Publishing Company, Inc., New York.

9] Schank, R. C., Lebowitz, M. and Birnbaum, L. (1980). An Integrated Understander. In American Journal of Computational Linguistics, vol. 6 no. 1, January-March 1980.

10] Wilensky, R. and Arens, Y. (1980a). PHRAN: A Knowledge-Based Approach to Natural Language Analysis. Berkeley Electronic Research Laboratory Memorandum No. UCB/ERL/M80/34.

11] Wilensky, R. and Arens, Y. (1980b). PHRAN - A Phrasal Natural Language Understander. In ACL 80: Proceedings of the Eighteenth Annual Meeting of the Association for Computational Linguistics. Philadelphia, Pennsylvania.