A parser analyzer of empirical design for question-answering*

by ABRAHAM S. BEN DAVID
Xerox Corporation
Rochester, New York

ABSTRACT

Over the last three years, work of an empirical nature has been carried out on the design of a natural-language question-answering system. In the present paper, the parser-analyzer of version 3.0 of the QUANSY (QUestion ANswer SYstem) system is discussed. The parser-analyzer is composed of a set of algorithms, each of which is assigned a particular set of tasks. The parser-analyzer operates in a multi-hypothesis structure. The whole system is programmed in FORTRAN and the sample run times are for a run on a CDC 6400.

INTRODUCTION

Research into natural language systems of all types has been going on for years.¹⁻² Even with all this effort, little headway has been made into the design and implementation of an efficient natural-language, question-answering system. The systems that have been implemented³⁻⁶ consume large amounts of computer time and memory and are of limited application. In most cases, previous attempts have utilized various linguistic theories in the design of their parsers. The question that should be posed is, "Is a theory of language explaining to humans how language works appropriate for machine analysis of language?"

The present research represents an empirical study into the question, "Can a model of language analysis specifically for machine analysis be developed?" The model developed for version 3.0 of QUANSY (QUestion ANswer SYstem) has been entirely of empirical design. Any idea, vis-à-vis language analysis, was actually tested by at least partial implementation, incorporated into the model if successful, modified—if possible—until successful, or discarded. Due to this empirical approach, work has progressed slowly, though "wrong paths" have usually been avoided. The results of the present version are detailed enough to demonstrate the potential of the model, though substantial additional work is necessary. After giving a sample QUANSY dialog, the paper will deal with the actual nature of the parser-analyzer of the system, and then with some concluding remarks concerning the import of this work.

SAMPLE SYSTEM OPERATION

The following sample operation is extracted from a larger interaction.³ All of the QUANSY requests and responses are indented and all capitalized. User replies are non-indented. Two performance measurements will be supplied, PT-parse times in CP seconds (these are times for the process of the parser-analyzer, as reported on in this paper) + TO total time in CP seconds (including physical input of material, dictionary search, memory modification and—for questions—formulation of responses and actual generation).

A mosaic is a picture that is made of colored stones. The first mosaics were probably made from slabs of alabaster which is a stone that is soft enough to cut. Parts of pictures were carved on the slabs and painted. These tiles or slabs were then put together to form a large picture. These large pictures were used to decorate the palaces and temple walls of kings in the Near East. The artists used the pictures to tell stories

I AM ANALYZING YOUR LAST 5 SENTENCES.
I'LL BE RIGHT BACK FOR MORE.

About the kings and what they did. -- --)

* This work has been supported in part by NSF Grants NSF-GN-40000 and NSF Grant No. SIS7-09822.
I AM READY FOR YOUR QUESTION
END IT WITH A PERIOD

Were mosaics used to decorate temples?
I HAVE CONDUCTED THE BASIC SEARCH WITHOUT SUCCESS.
DO YOU WANT ME TO CONTINUE THE SEARCH (YES OR NO).

Yes
LARGE PICTURES WERE USED TO DECORATE THE KINGS TEMPLE WALLS.**
(DO YOU WANT TO ASK ANOTHER QUESTION.

Did the mosaics tell stories about kings?
—I AM ANALYZING YOUR RESPONSE FOR CONTENT.—
I HAVE CONDUCTED, etc.

Yes
THE PICTURES TELLED† STORIES ON THE KINGS.

What were the earliest mosaics made from?
—I AM ANALYZING YOUR RESPONSE FOR CONTENT.—
I HAVE CONDUCTED, etc.

Yes
PICTURE PARTS WERE CARVED ON ALABASTER SLABS.

DISCUSSION OF SAMPLE SYSTEM OPERATION

The emphasis of this paper will be a discussion of the operation of the parser-analyzer (full details of other operations are available in Ben David†). In declarative mode, the natural language input is analyzed and formalized to the memory structure of this system. In the interrogatory mode, the natural language request is analyzed by the same procedures, formalized to the memory structure as a satisfy formulation rather than a modify formulation. Depending on the success of the satisfy formulation, either: (a) an answer is produced from the memory structure, or (b) a query of whether the system should continue search (something has been satisfied but more searching is necessary), or (c) a negative response is issued. A positive response to (b) results in generation of several variant satisfy formulations and a reiteration of the search.

PARSER-ANALYZER

The description of the parser-analyzer to be presented in this paper is intended to be complete enough to facilitate an in-depth comprehension of the system's operation and of how it differs from previous efforts. Due to space (and time) limitations (and in some cases unavailability of detailed documentation of the inner workers of other systems' operation) specific points of contrast to other systems cannot be given here (for overviews of other systems, see References 1, 2, 3 and 7).
The process about to be detailed has its "roots" in the first version of QUANSY itself and before that in the front-end of the LEADER retrieval system, see Hillman. However, the present process differs substantially from that of the previous versions. The explanation of the parser-analyzer will be presented as if the parser-analyzer were dealing with a particular sentence anywhere in text. It is not possible to simply start the description of the process; rather, there is an existing environment that must be described. At the very beginning of the analysis of a particular sentence (or throughput unit as described in Ben David) there exists a temporary knowledge structure. This structure contains whatever information has been just previously analyzed and is grouped as a cohesive memory unit. This structure is already linked (see Reference 7), but has not yet been entered in the regular memory of the system. This is because it is expected that additional information may be added in the next few input sentences which will directly affect this structure. The decision-process about whether or not the present sentence relates to this knowledge structure is described in Reference 7.

MANGRAM (Manage Grammatical Analysis)

This routine has no linguistic rules; it is strictly the controller of the grammatical analysis. The analysis is a succession of applications of linguistically oriented routines which are called by MANGRAM in the sequence HYPSTRC, SUBDETR, VRBDETR, OBJDETR. At the end of the sequence, MANGRAM checks the status of the analysis. If analysis is complete, it terminates operation; otherwise it performs the sequence again until analysis is complete. However, analysis is not continued indefinitely in this manner. If the number of sequences necessary to analyze a particular sentence becomes too great, MANGRAM will terminate the analysis and issue an error diagnostic.

One potential (and very important) topic of future research is the design of an efficient default mechanism in a "looping" situation. Because of the complexity of natural language, many things can go wrong in an analysis procedure which result in endless "looping" in an unsuccessful effort to find the best solution. A default procedure could be designed to settle for any solution that appeared at all reasonable and not hope for some ultimate or best solution. It should be noted that there is a need for many default procedures in a natural language analysis and this one is just the main one.

HYPSTRC (Hypothesize Structure)

The idea for the present overall approach to the first part of the analysis is derived from some of the ideas discussed by Ulric Neisser, in his book Cognitive Psychology. Neisser* says "We deal with the sentences we hear by reformulating them for ourselves; we grasp their structure with the same apparatus that structures our own utterances."

From the collection of the Computer History Museum (www.computerhistory.org)
ing a more detailed second hypothesis and passing the more detailed second hypothesis back to the system for verification. HYPSTRC makes a more detailed specification of the subject than of the predicate (this so as to make the determination of the beginning of the predicate as certain as possible). The verb phrase is grouped approximately (more or less certainly depending on the amount of ambiguity) and the rest of the present clause is lumped together into the "object"—subject to later processing.

(1) First time operations

The first time through for a particular through-put unit, usually a sentence, various initializations are performed—the subject, object, verb, conjunction, etc., pointers are all zeroed out. The routine SCAN is called to do the initial hypothesis. The initial clause boundaries, according to SCAN’s hypothesis, are picked up and the first-time operations are complete (more detail on SCAN’s operation is contained in the next few pages). If in the initial boundaries the first word is detected as a conjunction, then the routine CONJFST is called (details following).

(2) Other times operations

On successive times through HYPSTRC for the same throughput unit, there are similar initializations to the first time through, though necessary information must be saved.

Consider the sentences:

The boy is eating the cake that the girl baked. (1)

The boy that baked the cake is eating. (2)

On the second time through for sentence (1), the object of the first clause “the cake” should be saved, while for sentence (2) the subject of the first clause “the boy” should be saved. After the decision about what to save is made (based on “start” and “embed” pointers set in the previous run-through), all pointers are initialized accordingly and the next clause boundaries are picked up from SCAN’s hypothesis. (One very important initialization is the setting of the embed pointer to negative.)

(3) Find subject

To determine the subject of the clause, HYPSTRC proceeds word by word through the clause (as specified by the boundaries) looking for a verb, preposition, or determiner and ignoring other categories. In the process, it specifies the potential subject and distinguishes potential phrases in the subject. The subject is defined as all those phrases before the verb (including noun phrases, prepositional phrases, infinitive phrases, etc.). All words ignored are added to the present phrase under consideration.

(i) Verb

If a verb is found, is it auxiliary verb (form of to be, to have, to do, modal)? If this is auxiliary, assume that it signifies end of subject and beginning of verb phrase, exit this section. If not, note its location and continue. However, if there has been a previous verb noted thusly, retain the last if both are the same tense, retain the first if it is past tense, and retain the second (i.e., the one presently under consideration) if it is in present tense.

(ii) Preposition

If this is the first word in clause, continue (no operation). If not first word, it is “of”? If so, add it to present phrase (i.e., ignore it). For any other preposition, begin new phrase (but stay in subject—this section).

(iii) Determiner

If this is first word in clause, continue (i.e., ignore it). If the immediately preceding word is a preposition or determiner (i.e., “from the” or “the few”), ignore it. (The decision of what to do with this phrase was made in consideration of the last word.) If none of these circumstances apply, suspect that this determiner
might be the beginning of the object. If a potential verb has been detected (non-auxiliary noted in section on verbs, above), accept it as the beginning of verb phrase and exit this section.

It is altogether possible that the present boundary of the clause will be reached before this section is exited (i.e., either from the verb or determiner operations of the section “Find Subject”). The first question is whether there has been a potential verb (as noted in section on verbs). If there has been, take that verb as the beginning of the verb phrase and go on to the next section. If there hasn’t been a potential verb, is this the end of a throughput unit? If this is not the end of the throughput unit, what is the nature of the boundary (usually the boundary will be a conjunction but not always)? If the boundary is a conjunction of subcategory 3 or less, and the following word (after the conjunction) is not a determiner, ignore this boundary, set clause as up to next boundary, and continue processing from the beginning of this section (examples like “The boys and the girls” or “The red and black ball”). If the following word is a determiner, the boundary signals a new phrase: set pointers appropriately, pick up next boundary and continue processing from the beginning of this section.

If the boundary is a conjunction of subcategory 4 or greater, it signals the end of present phrase and embedded situation. Turn on embedded pointer (pointing to this location). Pick up next boundary and continue processing from the beginning of this section (as in example (2) above).

Not discussed in the description of this section is the handling of locational phrases and gerund phrases. If the boundary reached is one of these (they are detected and grouped by SCAN—details following), it is entered as the type of phrase indicated, almost transparently with regards to the above section.

In the aforementioned example (1), the processing in this section would have reached “is”, determined it as the beginning of the verb phrase and “the boy” as subject and processing goes on to the next section. For example (2), “that” is reached as the boundary, no potential verb has been detected and since the subcategory value of “that” is “4” the embedding pointer is turned on. The next boundary is picked up, to the end of the sentence, and processing begins again. “Is” is detected as the beginning of the verb phrase and processing goes on to the next section.

(4) Find verb phrase

This section assumes the first word in the verb phrase has been found. It groups all adverbs and verbs into the verb phrase until it reaches a verb participle or a word not categorized as either a verb or adverb (the verb participle is grouped into the verb phrase, non-adverbs or non-verbs are not). If the first verb detected was an auxiliary and that verb was the first in the sentence, the routine AUXFST is called. After checking if the AUXFST was successful, operation is returned to section three, “find subject.”

(5) Define object boundaries

The object is defined as everything from the end of the verb phrase to the present boundary. It is very possible that there is nothing between the end of the verb phrase and the boundary for this clause. If the boundary is the end of the throughput unit, then the next operation is the sufficiency check, otherwise the next step is section 6.

(6) Determine what to do

There are two distinct boundaries that can occur, conjunction of subcategory value 3 or less or conjunction of subcategory value 4 or greater.**

(i) Conjunction of subcategory value 3 or less.

Now there are two alternatives; either there has been a potential object or there hasn’t been. If there has been an object, the portion between the present boundary and the next boundary is scanned for a verb. If one is found, and it is the first word in the portion, a check is made to see if it is the same type as the last entry in the verb phrase (isolated by section 4 above). If it is, it is considered another verb phrase. For instance, in:

Parts of pictures were carved on the slabs and painted.

(3)

“Parts of pictures” is the subject, “were carved” is the verb phrase, “on the slabs” is the object (in the sense of object as defined in section 5—Define object boundaries), and “painted” would be detected by the above operation as an additional verb phrase.

If the verb is not the same type as the last entry in the verb phrase, this section tries to modify the category of this “verb” to noun and then ignores this word. If the verb is not the first in this portion of text, then is it an auxiliary? If so, this is the start for the next run-through. Add all words from one before auxiliary to object and terminate. If present clause is not embedded, this is a plain conjunctive situation; do not modify anything and terminate. If this verb is a non-auxiliary, has there been a previous potential verb? If so, process this second one like auxiliary (default situation). If there hasn’t been another potential verb, note this as one (both in the above circumstance and the following one, the situation is not clear and in effect, a fuzzy hypothesis is made and OBJDETR will have to deal with the problem).

At the termination of the scan, first check if any potential verbs have occurred; if so default out performing auxiliary operation specified above. If there has been no potential verb, check the next boundary. If it is conjunction of value 3 or less, add this whole portion to...
The boy is eating the cake and the pie that had been bought was being eaten by the child. (4)

When reaching section 6, “the boy is eating the cake” has been processed. The portion “the pie” is being considered. Since “that” is the next conjunction, it is not at all clear whether “the pie” goes with the first clause or the second. Therefore, “and” is noted as an ambiguous path and nothing else is done—no modification to the clause. HYPSTRC is terminated. If there hasn’t been an object and the word following conjunction is not a verb or adverb, set this point as start for next pass through and terminate. If the word following conjunction is a verb or adverb, assume conjunctive verb. Reset clause boundaries to include this portion and continue processing from section 4, “find verb phrase.”

(ii) Conjunction of subcategory value 4 or more

First turn embed pointer on. Is the conjunction “than”? If so, set “than” pointer and move modifier relating to “than” into verb phrase (i.e., bigger than, smaller than, more than, etc.). Set start pointer and terminate.

(7) Check Sufficiency

The first part of the check involves the value of the embed pointer. If it is less than or equal to zero, there is no problem and operation is terminated (if the embedding pointer is zero, then this is a simple sentence; if it is negative, then there has been enough information). If the embedding pointer is set, then the object grouping must be scanned for a potential verb. If a verb is found, is it an auxiliary? If so, modify object boundaries to the verb before auxiliary and return. If it is not an auxiliary, note it and continue scan. If at the end of the scan there has been a potential verb, make modifications, allow OBJDETR to complete determination. If no potential verb has been found, check for ambiguous paths. Has there been one? If not, issue error diagnostic and terminate; if there has been one—remove it and restart operation from first operation (call SCANMOD for removing boundary). For example, in (2) above, the verb phrase would be found as “baked” and the object as “the cake is eating.” When this section was called, the embed pointer would be set and the end of the through-put unit reached. In scanning the object, “is” is found, the object is modified to “the cake” (and the start pointer is modified for next time through routine). In the sentence:

The boy is eating the cake and the pie that had been bought. (5)

the first time through, “The boy is eating the cake” would have been set as the first clause. The second time through, “the pie” would be set as subject, “had been bought” as the verb phrase, and then the sufficiency check would determine a lack of sufficiency. Backtracking to the previously noted ambiguity, “and”, this path would be ruled out and processing restarted. In modified operation, “the boy is eating the cake and the pie” would be defined as the first clause, “The cake and the pie had been bought” as the second clause. (One important note—the problem of multiple ambiguous paths of this nature is not dealt with in this version except by successively backtracking).

SCAN

SCAN goes through a sentence word by word looking for a few central words which are:

(1) pronouns
(2) prepositions,
(3) conjunctions.

As the first part of the hypothesis analysis procedure, this routing has particular importance and is the most experimental. Its operation and justification for such will be described in some detail.

(1) Pronouns

The system distinguishes two main modes, question mode—when the user is querying the system, and declarative mode—when the system is analyzing declarative text. Pronouns are very important, particularly when the pronoun is first or second person and the system is in question mode. Very often the pronoun will be part of what can be referred to as the “Question-frame” or “Question-sign”—for example “What do you have about —”, “I want to know (why)”, etc. These occurrences are easily recognized and not passed on to the system except as question marks. When a question like this appears in declarative mode—i.e., as part of a text—for example “Joe asked ‘What do you have on ice cream’?”, then the “Question-frame” operation is not performed.

If a pronoun is detected, is this question mode or declarative mode? In declarative mode, ignore it. In question mode, set up the “question-frame” boundaries. Check the next three words. Is there a preposition? If so, set boundary to the preposition. If the second word following is also a preposition, set boundary to that preposition. If there isn’t a preposition in the first three words following the pronoun, don’t move this boundary at all. Set back boundary from previous conjunction, if there was one, or from the beginning of the sentence. Mark these boundaries as portion to be dropped.

(2) Prepositions

There are certain phrases which have well defined forms and which play important roles in a sentence (actually, as
will be seen shortly, what is being discussed here is not so much a phrase as a particular type of grouping of words which can include several prepositional phrases plus non-prepositional phrases. However, the unifying aspect is that this grouping of words relates a particular bit of information, of varying detail and exactness, with respect to location of two or more objects with regard to each other. In dealing with such phrases in this version, work has focused on the locational phrase. This type of phrase is among the most common of such phrases and appears in many forms; detection and isolation of these phrases has been through the prepositions commonly used with them. Some of the forms of these phrases are:

- south
- south of Boston
- to the south of Boston
- miles south of Boston
- miles to the south of Boston
- one hundred miles south of Boston
- one hundred miles to the south of Boston
- one hundred and fifty one miles south of the city of Boston

The same basic form appears in phrases like:

- right of the house,
- to the right of the house,
- etc.

In addition to its common appearance, it allows for some interesting inference experimentation and was therefore chosen for this version's work.

In order to detect the locational phrase, SCAN looks for the prepositions "of", "on", "to", and "from". It is certainly possible to generate a locational phrase without one of these prepositions (an example was given above—"south (of Boston)"), however, usually one or more of these prepositions will appear in the locational (in the case where none of these prepositions appear, the locational will be ignored—this is even though that potentially important information may be lost, for example "One hundred miles south is Philadelphia"). If the preposition, "of" is detected first, then we assume that this is the only preposition, otherwise we would have detected one of the others first. (Of course, there are locationals which have additional prepositions in their last part—"south of the City of Boston", "south of the highest point in the Rockies," etc.—the first "of" is of prime interest. If the grouping under consideration is considered a locational, then the problem of determining how far it extends is a separate problem from determining if it is a locational. If one of the other prepositions is detected first, then the routine looks for "of," which might not occur. In either case, a direction noun is critical; if one is found, then the question is to determine what the specific boundaries of the locational phrase are. Sometimes this decision is easy because the locational will be marked with commas; other times it is a process of preceding backwards and forwards from the detected prepositions looking for numeric and measure words in one direction and location specifics in the other. (Location specifics include names of places, location words—city, town, etc.) Once the boundaries of the locational are determined, it is treated as a unit. More than one locational can appear at one time—for instance in a compound locational. (For example "New York is one hundred miles north of Philadelphia and ninety miles east of Bethlehem.")

(3) Conjunction

This was the first consideration of SCAN. Originally, SCAN dealt only with conjunctions, and in a very simple manner. It soon became evident that additional processing routines were necessary in SCAN. First of all, several words which act like conjunctions but which are usually adverbs (i.e., "now"), especially when these words occur next to other conjunctions, can be modified immediately to save valuable processing time. (Actually, "now" is usually a modifier; however, it is sometimes used alone and the system must be ready. For instance—"Now (that) they have gone, the plans have arrived."—Usually "that" would appear with now; however, "that" will be left out sometimes, with "now" assuming a conjunction role.) Second, SCAN attempts some preliminary work with comparatives, focusing on the occurrence of the conjunction "than" in examples like—"bigger than," "smaller than," etc. There has been much work on conjunctions detailed in the literature (particularly in comparatives), however, because of the general emphasis of the present work on a question-answering system, most of the previous work was not judged suitable to the present effort.

SCANMOD simply takes direction from HYPSTR and with regard to turning off a specific boundary which was set in the first call to SCAN from HYPSTR. It also sets an index value so that when SCAN is called by HYPSTR (because processing is reinitiated after a call to SCANMOD and SCAN will be called in first run processing by HYPSTR) nothing will be done by SCAN.

CONJFST-AUXFST

This routine has the responsibility of undoing permutations due to the interrogatory transformation. This transformation is detected either by the question pronoun as the first word ("what", "which", etc.) or by a lead auxiliary verb ("is", "did", "have", etc.) These cases are combined because they require similar actions. (First the details of the operations of CONJFST will be discussed and then the combined operation of CONJFST and AUXFST.)

(1) Conjunction first word

The various question pronouns are also used as conjunctions. For example, "What is the boy eating?" and "The
boy is eating what he can." While their usages are different, their senses are similar. In the system's operation, all of the question pronouns are categorized first as conjunctions (of subcategory 4 or greater) and secondly as question pronouns. When a conjunction is detected as the first word of a sentence (by HYPSTRC), this routine is called and it attempts to modify the category from conjunction to question pronoun. If it cannot modify the conjunction (because this conjunction is not a question pronoun), it does nothing except note the conjunction.

If the conjunction has been converted to a question-pronoun, CONJFST then tries to determine the nature of the question. There are those questions where the only task is noting what the question-pronoun is—"What boy is coming?" For others, more work is necessary—"What is the mileage from Chicago to San Francisco?" This is a very specific "what question—"What is the mileage" is the same as "How far is it," "How many miles is," etc. As much as possible about a particular question must be determined as early as possible in order to speed analysis and increase the accuracy of the response. The boundaries of the particular "question frame" are determined and subtracted from the input (ignored by the rest of the system).

Even if the first word (or group of words) is determined to be a question-frame, it is not certain that a permutation is also present. For example, "What boys are coming tomorrow?" and "What is eating the cheese?" The task for a permutation involves checking for a split verb phrase. This necessitates some auxiliary verb plus a participle. If the auxiliary does not follow immediately after the question frame, suspect a double permutation, for instance "How much gas does tank five contain?" "Tank five" and "does" must be interchanged and then, "How much gas" must be moved to the end of the question. When a double permutation seems likely, the second permutation section is called.

(2) Auxiliary first word

When an auxiliary occurs as the first word of the sentence, the first part of this routine, CONJFST, is skipped. However, just as in the last part of CONJFST, certain predictions are made vis-à-vis what permutations are expected. In this circumstance, a single permutation is expected.

(3) Permutation operation—one permutation (entry for AUXFST and for auxiliary after question frame from CONJFST).

In the permutation check, AUXFST scans for verbs, determiners, and prepositions.

(i) verb found

Is this verb an auxiliary or a verb participle? If so, does it follow the first auxiliary immediately? If so, no permutation—issue error diagnostic and return. If this verb does not follow first auxiliary immediately, set move unit from word after first auxiliary to word before present word. The entry point (i.e., where this move unit belongs) is before first auxiliary. For example, in "Are the boys eating cereal," "the boys" will be move unit and it would be moved before "Are" resulting in "The boys are eating cereal."

(ii) determiners

Is this first after auxiliary—if so skip it. If the word preceding this determiner was a preposition or another determiner—skip it (i.e., the decision about whether to continue the scan or not was made with the previous word). Otherwise, set move unit from word after first auxiliary to word before this one. The entry point is location before first auxiliary. For example, in "Are the boys the ones?", "The boys" is the move unit, the rearrangement results in "the boys are the ones."

(iii) prepositions

Is this first word after auxiliary—if so skip it. Has there been a preposition already? If not, note this one and continue. If there has been one, is this one "of"? If so skip it, otherwise note it in place of the previous preposition. If the end of the scan is reached without a "move unit" established, check if there has been a preposition. If so, set move unit from word after first auxiliary to word before noted preposition and entry point in front of first auxiliary. For example "Are the boys from New York?" becomes "The boys are from New York." Note that "Are the boys from New York coming?" would become "The boys from New York are coming."

(4) Permutation operation—two permutations (this section has only been developed to handle one type of situation—more work will follow).

When this section is called, it has no information of a first auxiliary. It first scans for the first auxiliary, notes it and then scans for a second auxiliary or participle (which can also be an auxiliary). If the auxiliary and the second verb are next to each other, no rearrangement is performed. The first move unit is from the word after the first auxiliary to word before second verb—entry point before first auxiliary, second move unit is from first word after question frame to word before first auxiliary—entry point after second verb. For example—"How much gas does tank five contain?"

SUBDETR—OBJDETR

These routines have certain basic similarities, but while the break-up of the constituent phrases in the subject has
been taken care of by HYPSTRC to assist in accurate determination of the beginning of the verb phrase, none of the constituent phrases of the object have been isolated (it is important to remember that in referring to the subject and object, the reference is to all the various phrases before and after the verb phrase, as discussed earlier). There are various reasons for this differentiation; most notable is that until the verb phrase is fully determined—it is not clear where the object begins.

Both of these routines are concerned with filling the array PHRVALS. The following locations exist in this array:

1. Preposition
2. Determiner
3. Number—first location
4. Number—last location
5. Modifiers—first location
6. Modifiers—last location
7. Noun—main noun of phrase
8. Conjunction
9. Infinitive
10. Adverbs

When these routines have completed operation of a particular phrase, this array (sometimes in multiples) is passed to PHRCLOS (discussed later). However, if pronouns are detected, the routines CASEPRI or CASEPR2 are called (1 for subject, 2 for object) and PHRCLOS is not called.

The operation is pretty straightforward setting up the array. The exceptions are as follows

(1) Preposition

Prepositions are handled basically the same, except for the following difference in these routines.

(ii) OBJDETR

OBJDETR treats "of" basically the same as SUBDETR; however, any other preposition encountered is considered the basis for a phrase break unless it is the first word in the object. If it is "from", check for "to" after "from" (i.e., "from Chicago to New York"). In case of a phrase break, the routine PHRCLOS is called.

(2) Conjunctions

Conjunctions have the same impact in OBJDETR and in SUBDETR, except that (just as with prepositions) in SUBDETR there is less expectation that the conjunction might signify a new phrase, as the calculations on that point would have been done in HYPSTRC. In both routines the location of the conjunction is noted in PHRVALS (8) and the phrase indicator is incremented—new information occurring will be entered in another level of PHRVALS, not interfering with the previous information entered therein.

PHRCLOS

PHRCLOS looks at the structure of each phrase (as sent by SUBDETR and OBJDETR) in the array PHRVALS and completely categorizes the phrase. If there is a determiner, it picks up the determiner code and stores it for the phrase. It determines whether a phrase with a conjunction is two phrases or one phrase with a compound modifier. If there are two phrases, does the same determiner and modification apply to both phrases? Is there a number present? If so evaluate it. Is there a preposition? If so, what kind of prepositional phrase is this (CASEVAL)? Based on the information it gathers, it decides whether the phrase is singular or plural.

CASEVAL

This routine is called to calculate the nature of a prepositional phrase. In order to deal with the great amount of information present in nouns, this routine employs decision tables. Using the information contained in the preposition and the noun, this routine determines what a particular phrase specifies about a sentence.

CASEPRI-CASEPR2

This routine (the different names just represent different entry points) calculates the antecedents of pronouns using positional, number, and type (human, non-human) cues.

The routine has two knowledge structures to consider. One is the one presently being built for this sentence (a through-put unit). This knowledge structure is not linked

...
and will only be completed when the whole sentence has been processed; however, it is available for checking by this routine. For instance in the sentence, "The artists used the pictures to tell stories about the kings and what they did", from the beginning of the sentence to "and" would be processed as the first clause and "what they did" as the second. This is an ambiguous situation and the routine would settle for "the artists" as the referent of "they" (based on positional cue). The other knowledge structure that this routine has access to is the one this sentence fits into, provided there is one. It is entirely possible that neither of these structures is available (i.e., at the beginning of a particular text). If neither of the knowledge structures contains any information, the routine will issue diagnostics unless the referent of the pronoun could be following (i.e., "It is the boy who is responsible.")

**VRBDETR**

VRBDETR isolates the verb phrase, determines the tense and transformations, and takes care of adverbial modifications of verb whether negative or positive. This routine makes extensive use of decision tables to facilitate fast operation. It is distinctly possible that the determination of the verb phrase made by HYPSTRC will be inaccurate—usually overlong. VRBDETR will modify the object boundaries by moving any information it doesn’t need into the object. This happens particularly when new words are introduced or a word is used without a determiner preceding it (i.e., as a predicate adjective, as in “The boy is sick” where “sick” is treated as a noun with associated subcategories—see previous discussion).

**CONCLUSION**

The parser-analyzer for QUANSY 3.0, as presented in this paper is of straightforward syntactic and semantic nature (though predominantly syntactic). As it stands right now, it is able to handle levels of English up to a fourth-grade level at relatively high speeds. Most important, the parser-analyzer, as presented here in depth, can be seen to be distinctly different from other approaches to natural-language analysis. With the introduction of additional semantic capabilities, the system should be able to achieve substantially higher levels of performance.

**REFERENCES**