Paraphrasing Questions Using Given and New Information[1]

Kathleen R. McKeown

Computer Science Department
Columbia University
New York, NY 10027

The design and implementation of a paraphrase component for a natural language question-answering system (CO-OP) is presented. The component is used to produce a paraphrase of a user's question to the system, which is presented to the user before the question is evaluated and answered. A major point made is the role of given and new information in formulating a paraphrase that differs in a meaningful way from the user's question. A description is also given of the transformational grammar that is used by the paraphraser.

1. Introduction

In a natural language interface to a data base query system, a paraphraser can be used to ensure that the system has correctly understood the user. Such a paraphraser has been developed as part of the CO-OP system (Kaplan 1979). In CO-OP, an internal representation of the user's question is passed to the paraphraser, which then generates a new version of the question for the user. Upon seeing the paraphrase, the user has the option of rephrasing her/his question before the system attempts to answer it. Thus, if the question was not interpreted correctly, the error can be caught before a possibly lengthy search of the data base is initiated. Furthermore, the user is assured that the answer she/he receives is an answer to the question asked and not to a deviant version of it.

The idea of using a paraphraser in the above way is not new. To date, other systems have used canned templates to form paraphrases, filling in empty slots in the pattern with information from the user's question (Waltz 1978, Codd 1978). In CO-OP, a transformational grammar is used to generate the paraphrase from an internal representation of the question. Moreover, the CO-OP paraphraser generates a question that differs in a meaningful way from the original question. It makes use of a distinction between given and new information to indicate to the user the existential presuppositions made in her/his question.

2. Overview of the CO-OP System

The CO-OP system is aimed at infrequent users of data base query systems. These casual users are likely to be unfamiliar with computer systems and unwilling to invest the time needed to learn a formal query language. Being able to converse naturally in English enables such persons to tap the information available in a data base.

In order to allow the question-answering process to proceed naturally, CO-OP follows some of the "cooperative principles" of conversation (Grice 1975). In particular, the system attempts to find meaningful answers to failed questions by addressing any incorrect assumptions the questioner may have made in her/his question. When the direct response to a question would be simply "no" or "none", CO-OP gives a more informative response by correcting the questioner's mistaken assumptions.

The false assumptions that CO-OP corrects are the existential presuppositions of the questions.[2] Since these presuppositions can be computed from the surface structure of the question, a large store of semantic knowledge for inferencing purposes is not needed.

[1] This work was carried out in the Department of Computer and Information Science, The University of Pennsylvania. It was partially supported by an IBM Fellowship, and by NSF grants MCS 78-08401 and MCS 79-19171.

[2] For example, in the question "Which users work on projects sponsored by NASA?", the speaker makes the existential presupposition that there are projects sponsored by NASA.

0362-613X/83/010001-10$03.00

In fact, a lexicon and data base schema are the only items that contain domain-specific information. Consequently, the CO-OP system is a portable one; a change of data base requires that only these two knowledge sources be modified.

3. The CO-OP Paraphraser

CO-OP's paraphraser provides the only means of error checking for the casual user. If the user is familiar with the system, she/he can ask to have the intermediate results printed, in which case the parser's output and the formal data base query will be shown. The naive user, however, is unlikely to understand these results. It is for this reason that the paraphraser was designed to respond in English.

The use of English to paraphrase queries creates several problems. The first is that natural language is inherently ambiguous. A paraphrase must clarify the system's interpretation of possible ambiguous phrases in the user's question without introducing additional ambiguity.

One particular type of ambiguity that a paraphraser must clarify and avoid re-introducing is caused by the linear nature of sentences. A modifying relative clause, for example, frequently cannot be placed directly after the noun phrase it modifies. In such cases, the semantics of the sentence may indicate the correct choice of modified noun phrase, but occasionally the sentence may be genuinely ambiguous. For example, question (A) below has two interpretations, both equally plausible. The speaker could be referring to books dating from the '60s or to computers dating from the '60s.

(A) Which students read books on computers dating from the '60s?

A second problem in paraphrasing English queries is the possibility of generating the exact question that was originally asked. If a grammar were developed to simply generate English from an underlying representation of the question, this possibility could be realized. Instead, a method must be devised that can determine how the phrasing should differ from the original.

The CO-OP paraphraser addresses both the problem of ambiguity and the rephrasing of the question. It makes the system's interpretation of the question explicit by breaking down the clauses of the question and reordering them depending upon their function in the sentence. Thus, question (A) above will result in either paraphrase (B) or (C), reflecting the interpretation the system has chosen.

(B) Assuming that there are books on computers (those computers date from the '60s), which students read those books?

(C) Assuming that there are books on computers (those books date from the '60s), which students read those books?

The method adopted generates a paraphrase that differs from the original except in cases where no relative clauses or prepositional phrases were used. It was formulated on the basis of a distinction between given and new information and indicates to the user the presuppositions she/he has made in the question (in the "assuming that" clause), while focusing her/his attention on the attributes of the class she/he is interested in.

4. Linguistic Background

As mentioned earlier, the lexicon and the data base are the sole sources of world knowledge for CO-OP. While this design increases CO-OP's portability, it means that little semantic information is available for the paraphraser's use. Contextual information is also limited since no running history or context is maintained for a user session in the current version. The input the paraphraser received from the parser is a syntactic parse tree of the question. Using this information, the paraphraser must construct a question that differs in phrasing from the original. The following question must therefore be addressed:

> What reasons are there for choosing one syntactic form of expression over another?

Some linguists maintain that word order is affected by functional roles elements play within the sentence.[3] Terminology used to describe the types of roles that can occur varies widely. Some of the distinctions that have been described include given/new, topic/comment, theme/rheme, and presupposition/focus. Definitions of these terms, however, are not consistent.[4]

Nevertheless, one influence on expression does appear to be the interaction of sentence content and the beliefs of the speaker concerning the knowledge of the listener. Some elements in the sentence function in conveying information the speaker assumes is present in the "consciousness" of the listener (Chafe 1976). This information is said to be contextually dependent, either by virtue of its presence in the preceding discourse or because it is part of the shared world knowledge of the dialog participants. In a question-answering system, shared world-knowledge

[3] Some other influences on syntactic expression are discussed in Morgan and Green 1973. They suggest that stylistic reasons, in addition to some of the functions discussed here, determine when different syntactic constructions are to be used. They point out, for example, that the passive tense is often used in academic prose to avoid identification of agent and to lend a scientific flavor to the text.

[4] For example, see Prince 1979 for a discussion of various usages of "given/new".

refers to information the speaker assumes is present in the data base. Information functioning in the role just described has been termed "given".

"New" labels all information in the sentence that is presented as not retrievable from context. In the declarative, elements functioning in asserting information that the listener is presumed not to know are called new. In the question, elements functioning in conveying what the speaker wants to know (i.e., what she/he doesn't know) represent information the speaker presumes the listener is not already aware of. Firbas 1974 identifies additional functions in the question. Of these, (ii) is used here to augment the interpretation of new information. He says (p. 31):

> (i) it indicates the want of knowledge on the part of the inquirer and appeals to the informant to satisfy this want.
>
> (ii) [a] it imparts knowledge to the informant in that it informs him what the inquirer is interested in (what is on her/his mind) and [b] from what particular angle the intimated want of knowledge is to be satisfied.

Although word order vis-a-vis these and related distinctions has been discussed in light of the declarative sentence, less has been said about the interrogative form. Halliday 1967 and Krizkova[5] are among the few to have analyzed the question. Despite the fact that they arrive at different conclusions,[6] the two follow similar lines of reasoning. Krizkova argues that both the wh-item of the wh-question and the finite verb (e.g., "do" or "be") of the yes/no question point to the new information to be disclosed in the response. These elements, she claims, are the only unknowns to the questioner. Halliday, in discussing the yes/no question, also argues that the finite verb is the only unknown. The polarity of the text is in question and the finite element indicates this.

In this paper the interpretation of the unknown elements in the question as dfined by Krizkova and Halliday is followed. The wh-items, in defining the questioner's lack of knowledge, act as new information. Firbas's analysis of the functions in questions is used to further elucidate the role of new information in questions. The remaining elements are given information. They represent information assumed by the questioner to be true of the data base domain. This labeling of information within the question will allow the construction of a natural paraphrase, avoiding ambiguity.

5. Formulation

Following the analysis described above, the CO-OP paraphraser breaks down questions into given and new information. More specifically, an input question is divided into three parts, of which (2) and (3) form the new information.

1. given information
2. lack of knowledge (ii[a] from Firbas above)
3. angle (ii[b] from Firbas above)

In terms of the question components, part (2) is indicated by the question with no subclauses[7] as it defines the lack of knowledge of the hearer. Part (3) is indicated by the direct and indirect modifiers of the interrogative words as they define the angle from which the question was asked. They identify the attributes of the missing information for the hearer. Part (1) is formed from the remaining clauses.

As an example, consider question (D):

(D) Which division of the computing facility works on projects using oceanography research?

Following the outline above, part (2) of the paraphrase will be the question minus the subclauses: "Which division works on projects?" Part (3), the modifiers of the interrogative words, will be "of the computing facility", which modifies "which division".[8] The remaining clause "projects using oceanography research" is considered given information. The three parts can then be assembled into a natural sequence:

(E) Assuming that there are projects using oceanography research, which division works on those projects? Look for a division of the computing facility.[9]

Information belonging to each of the three categories occurred in question (D). If one of these types of information is missing, the question will be presented minus the initial or concluding clauses. Only part (2) of the paraphrase will invariably occur. Note that this means that if there are no clauses in the original question corresponding to parts (1) and (2) (i.e., the question contains no relative clauses, prepositional phrases,

[5] Summary by Firbas 1974 of the untranslated article "The Interrogative Sentence and Some problems of the So-called Functional Sentence Perspective (Contextual Organization of the Sentence)," NASA Rec. 4, 1968.

[6] It should be noted that Halliday and Krizkova discuss the unknowns in the question in order to define the theme and rheme of a question. Although they agree about the unknowns for the questioner, they disagree about which elements function as theme and which function as rheme. A full discussion of their analysis and conclusions is given in McKeown 1979.

[7] Here, subclauses are defined as relative clauses, prepositional phrases, and adjectival phrases.

[8] Note that this phrase also identifies a presupposition of the questioner. For the paraphrase, however, its function to precisely specify what the questioner is interested in (which is new information for the hearer) is of greater importance.

[9] This example, as well as sample questions and paraphrases that follow, were taken from actual sessions with the paraphraser. Question (A) and its possible paraphrases (B) and (C) were not run on the system.

or adjectival phrases), the paraphrase may be the same as the original question.

If more than one clause occurs in a particular category, the question will be further splintered. Additional given information is parenthesized following the "assuming that ..." clause. Example (F) below illustrates the paraphrase for a question containing several clauses of given information and no clauses defining specific attributes of the missing information. Clauses containing information characterized by category (3) will be presented as separate sentences following the stripped-down question. (G) below demonstrates a paraphrase containing more than one clause of this type of information.

(F) Q: Which users work on projects in oceanography that are sponsored by NASA?
P: Assuming that there are projects in oceanography (those projects are sponsored by NASA), which users work on those projects?

(G) Q: Which programmers in superdivision 5000 from the ASD group are advised by Thomas Wirth?
P: Which programmers are advised by Thomas Wirth? Look for programmers in superdivision 5000. The programmers must be from the ASD group.

6. Implementation Overview

The paraphraser's first step in processing is to reform the parse tree it is given so that the main verb occurs as the root of the new tree. This is done to simplify the identification of given and new information in the parse. The tree is then divided into three separate trees reflecting the division of given and new information in the question. The design of the tree allows for a simple set of rules that flatten the tree. The final stage of processing in the paraphraser is translation. In the translation phase, labels in the parser's representation are translated into their corresponding words. During this process, necessary transformations of the grammar are performed upon the string.

6.1 The phrase structure tree

In its initial processing, the paraphraser transforms the parser's representation into one that is more convenient for generation purposes. The resultant structure is a tree that highlights certain syntactic features of the question. This initial processing gives the paraphraser some independence from the CO-OP system. Were the parser's representation changed or the component moved to a new system, only the initial processing phase would need to be modified.

The paraphraser's phrase structure tree uses the main verb of the question as the root node of the tree. The subject of the main verb is the root node of the left subtree, the object (if there is one) the root node of the right subtree. In the current system, the use of binary relations in the parser's representation[10] creates the illusion that every verb or preposition has a subject and object. The paraphraser's tree does allow for the representation of other constructions should the incoming language use them.

Note that the use of binary relations in the incoming parse tree to represent the verbs and prepositions of a sentence means that modifiers of verbs are represented as modifiers of their objects (and thus hang off the object in the paraphraser's reformed tree). While this is not the usual interpretation of questions using such constructions, it functions adequately for both CO-OP and the paraphraser as illustrated by a hypothetical paraphrase for such a question, shown below in (H):

(H) Q: Which programmers worked on oceanography projects in 1972?
P: Assuming that there were oceanography projects in 1972, which programmers worked on those projects?

Each of the paraphrase subtrees represents other clauses in the question. Both the subject and the object of the main verb will have a subtree for each other clause it participates in. If a noun in one of these clauses also participates in another clause in the sentence, it too will have subtrees.

As an example, consider the question: "Which active users advised by Thomas Wirth work on projects in area 3?" The phrase structure tree used in the paraphraser is shown in Figure 1. Since "work on" is identified as the main verb of the question by the parser, it will be the root node of the tree. "users" is root of the left subtree, "projects" of the right. Each noun participates in one other clause and therefore has one subtree. Modifiers are closely bound to the noun they modify and are treated as properties of the noun (i.e., each node in the tree that is modified has a property called "modifiers" whose value is any adjectival or noun modifier). In Figure 1, modifiers are shown as part of the node label for clarity. Subtree nodes (the leaves of Figure 1) have three pieces of information associated with them:

- the relation between the node and its parent,
- the noun phrase the node represents, and
- an indication of whether the node functions as subject or object in the clause.

[10] See Kaplan 1979 for a description of Meta Query Language, or MQL.

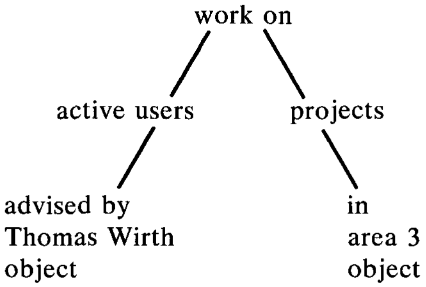
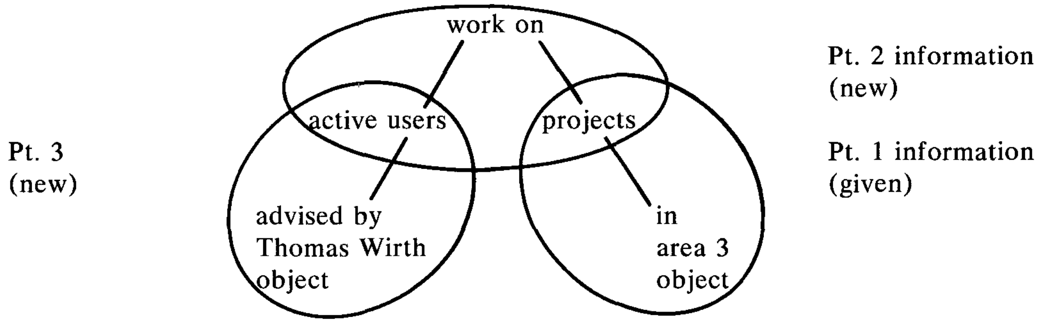


Figure 1.

6.2 Dividing the tree

The constructed tree is computationally suited for the three-part paraphrase. The tree is flattened after it has been divided into subtrees containing given information and the two types of new information. The splitting of the tree is accomplished by first extracting the topmost smallest portion of the tree containing the wh-item. At the very least, this will include the root node plus the left and right subtree root nodes. This portion of the tree is the stripped-down question. The clauses that define the particular aspect from which the question is asked are found by searching the left and right subtrees for the wh-item or questioned noun. The subtree whose root node is the wh-item contains these clauses. Note that this may be the entire left or right subtree or may only be a subtree of one of these. The remainder of the tree represents given information. Figure 2 illustrates this division for the previous example.



Q: Which active users advised by Thomas Wirth work on projects in area 3?

P: Assuming that there are projects in area 3, which active users work on these projects? Look for users advised by Thomas Wirth.

Figure 2.

6.3 Flattening

If the structure of the phrase structure tree is

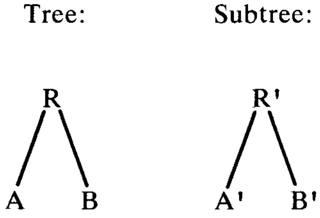


Figure 3.

with A the left subtree and B the right, then the following rules define the flattening process:

TREE → A R B

SUBTREE → R' A' B'

In other words, the top level of the tree (shown on the left in Figure 3) is linearized by an in-order traversal while each of its subtrees (shown on the right in Figure 3) is linearized by a pre-order traversal. In the example shown in Figure 2, part (2) of the tree corresponds to the top level of the tree and will undergo in-order linearization, and parts (1) and (3) are the subtrees, which will be linearized by a pre-order traversal. The use of two traversals to linearize the tree stems from the fact that different types of information are stored at nodes at different levels in the tree. As a node in a subtree has three pieces of information associated with it, one more rule is required to expand a node. A nóde consists of:

- ▶ arc-label
- ▶ set-label
- ▶ subject/object

where arc-label is the label of a binary relation in the input parse tree (i.e., a verb or preposition) and set-label is the label of a set in the input parse (i.e., noun phrase). The input parse is in MQL representation, which consists of sets and binary relations between them. Subject/object indicates whether the sub-node noun phrase functions as subject or object in the clause; it is used by the subject-aux transformation and does not apply to the expansion rule. In Figure 2, the leaves of the tree carry these three pieces of information. For example, the leftmost leave has arc-label *advised by*, set-label *Thomas Wirth*, and is labeled as the object of the relation. The following rule expands a subtree node:

NODE → ARC-LABEL SET-LABEL

The tree of given information is flattened first. It is part of the left or right subtree of the phrase structure tree and therefore is flattened by a pre-order traversal. It is during the flattening stage that the words "Assuming that there [be] ..." are inserted to introduce the clause of given information. "be" will agree with the subject of the clause. Following these rules, the tree of given information in Figure 2 would be flattened by a pre-order traversal yielding "projects in area #6" (R' A' arc-label set-label). After the "Assuming that" clause is inserted, this portion of the paraphrase is "Assuming that there be projects in area #6". If there is more than one clause, parentheses are inserted around the additional ones.

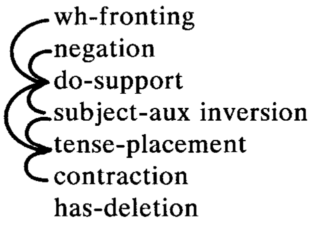
The tree representing the stripped-down question is flattened next, using the in-order traversal. Applying this process to Part (2) of the tree in Figure 2 yields the phrase "wh active users work on projects" (A R B). (In final processing stages, the correct demonstrative ("those" or "that") is selected to modify nouns already mentioned in the first part of the paraphrase.)

The tree that represents modifiers of the questions noun is linearized to follow these phrases. A preorder traversal of this portion of the tree in Figure 2 yields "users advised by Thomas Wirth" (R' A' arc-label set-label). Any modifiers of a noun (here, "active") are omitted in this part of the paraphrase if they have already been mentioned. The phrase "Look for" is inserted before the first clause of modifiers.

Two transformations are applied during the flattening process. They are wh-fronting and subject-aux inversion. Other transformations are applied following the flattening process to produce sentences in final grammatical form.

6.4 Transformations

The grammar used in the paraphrase is a transformational one. In addition to the basic flattening rules described above, the following transformations are used:



wh-fronting
negation
do-support
subject-aux inversion
tense-placement
contraction
has-deletion

The curved lines indicate the ordering restrictions. There are two connected groups of transformations. If wh-fronting applies, then so will do-support, subject-aux inversion, and tense-placement. The second group of transformations is invoked through the application of negation. It includes do-support, contraction, and tense-placement. Has-deletion is not affected by the absence or presence of other transformations. A description of the transformation rules follows. The rules used here are based on analyses described by Akmajian and Heny (1975) and by Cullicover (1976).

The rule for wh-fronting is specified as follows, where SD stands for structural description and SC, structural changes. Each rule is followed by an example input string and the string after it has undergone the transformation. The full tree for the string is not shown, but the string is labeled by markers in the SD.

SD:	X –	NP –	Y
	1	2	3
SC:	2+1	0	3

Input to rule:

$\overbrace{\text{programmers in division 5 past plur work on}}^{1}\quad\overbrace{\text{wh projects?}}^{2}$

Transformed input:

$\overbrace{\text{wh projects}}^{2}\quad\overbrace{\text{programmers in division 5 past plur work on?}}^{1}$

The first step in the implementation of wh-fronting is a search of the tree for the wh-item. A slightly different approach is used for paraphrasing than would be used if simply generating a question from the input parse. The difference occurs because in the original question the NP to be fronted may be the head noun of some relative clauses or prepositional phrases. If generating, these clauses would be fronted along with the head noun. Since the clauses of the original question are broken down for the paraphrase, it will never be the case when paraphrasing that the NP to be fronted also dominates relative clauses or prepositional phrases. For this reason, the applicability of wh-fronting is testing for and is applied in the flattening process of the stripped-down question. Note that the phrase markers (or categories) of each word are retained as the tree is flattened and thus the SD's can be matched against both the tree and its linearized version. If wh-fronting applies, only one word need be moved to the initial position.

The paraphraser is capable of generating English from the input as well as paraphrasing (see Section 7). When generation is being done, the applicability of wh-fronting is tested for immediately before flattening. If the transformation applies, the tree is split. The subtree of which the wh-item is the root is flattened separately from the remainder of the tree and is attached in fronted position to the string resulting from flattening the other part.

After wh-fronting has been applied, do-support is invoked. In CO-OP, the underlying representation of the question does not contain modals of auxiliary verbs. Thus, fronting the wh-item necessitates supplying an auxiliary. The following rule is used for do-support:

```
SD: NP - NP - tense - num - V - X
    1    2      3              4
SC: 1    2+do   3              4
condition:  1 dominates wh
```

Input to rule:

$\overbrace{\text{wh projects}}^{1}\ \overbrace{\text{programmers in division 5}}^{2}$

$\overbrace{\text{past plur work on?}}^{3}$

Transformed input:

$\overbrace{\text{wh projects}}^{1}\ \overbrace{\text{programmers in division 5 do}}^{2\,+\,\text{do}}$

$\overbrace{\text{past plur work on?}}^{3}$

Subject-aux inversion is activated immediately afterwards. Again, if wh-fronting is applied, subject-aux inversion will apply also. The rule is:

```
SD: NP - NP - AUX - X
    1    2     3    4
SC: 1    3+2   0    4
condition:  1 dominates wh
```

Input to rule:

$\overbrace{\text{wh projects}}^{1}\ \overbrace{\text{programmers in division 5}}^{2}\ \overbrace{\text{do}}^{3}$

$\overbrace{\text{past plur work on?}}^{4}$

Transformed input:

$\overbrace{\text{wh projects}}^{1}\ \overbrace{\text{do}}^{3}\ \overbrace{\text{programmers in division 5}}^{2}$

$\overbrace{\text{past plur work on?}}^{4}$

Tense-placement follows subject-aux inversion. Tense, number, and negation (if present) are attributes of all verbs in the parser's representation. When an auxiliary is generated, the tense, number, and negation are moved from the verb to the auxiliary. Formally:

```
SD: X - AUX - Y - tense-num (-no-) V - Z
    1    2    3      4             5   6
SC: 1   2+4   3      0             5   6
```

Input to rule:

$\overbrace{\text{Wh projects}}^{1}\ \overbrace{\text{do}}^{2}\ \overbrace{\text{programmers in division 5}}^{3}$

$\overbrace{\text{past plur}}^{4}\ \overbrace{\text{work on?}}^{5}$

Transformed input:

$\overbrace{\text{Wh projects}}^{1}\ \overbrace{\text{do}}^{2}\ \overbrace{\text{past plur}}^{4}$

$\overbrace{\text{programmers in division 5}}^{3}\ \overbrace{\text{work on?}}^{5}$

Some transformational analyses propose that wh-fronting and subject-aux inversion apply to the relative clause as well as the question. In the CO-OP paraphraser, the head-noun is properly positioned by the flattening process and wh-fronting need not be used. Subject-aux inversion, however, may be applicable. In cases where the head noun of the clause is not its subject, subject-aux inversion results in the proper order.

The rule for negation is tested during the translation phase of execution. It has been formalized as:

```
SD: X - tense-num-V - NP - Y
    1       2          3    4
SC: 1      2+no        3    4
condition:  4 marked as negative
```

Input to rule:

$\overbrace{\text{wh students}}^{1}\ \overbrace{\text{pres plur have}}^{2}\ \overbrace{\text{advisors?}}^{3}$

(advisors has property "neg")

Transformed input:

$\overbrace{\text{wh students}}^{1}\ \overbrace{\text{pres plur have}}^{2}\ \overbrace{\text{no}}^{+\ \text{no}}\ \overbrace{\text{advisors?}}^{3}$

In the CO-OP representation, an indication of negation is carried on the object of a binary relation (see Kaplan 1979). When generating an English representation of the question, it is possible in some cases to express negation as modification of the noun (see question (H) below). In all cases, however, negation can be indicated as part of the verb (see version (I) of question (H)). Therefore, when the object is marked as negative, the paraphraser moves the negation to become part of the verbal element.

(H) Which students have no advisors?
(I) Which students don't have advisors?

In English, the negative marker is attached to the auxiliary of the verbal element and, therefore, as was the case for questions, an auxiliary must be generated. Do-support is used. The rule for do-support after negation differs from the one used after wh-fronting. They are presented this way for clarity, but could have been combined into one rule.

```
SD: X - tense-num-V-no - Y
    1       2            3
SC: 1      do+2          3
```

Input to rule:

```
       1                2              3
┌───────────┐ ┌──────────────────┐ ┌─────────┐
 wh students   pres plur have no    advisors?
```

Transformed input:

```
       1        do +         2                3
┌───────────┐ ┌────┐ ┌──────────────────┐ ┌─────────┐
 wh students    do    pres plur have no    advisors?
```

Tense-placement, as described above, moves the tense, number, and negation from the verb to the auxiliary verb. The cycle of transformations invoked through application of negation is completed with the contraction transformation. The statement of the contraction transformation is:

```
SD: X - do+tense-num-V -no - Y
    1       2          3   4   5
SC: 1    #2+n't#       3   0   5
```

Input to rule:

```
       1              2          3     4      5
┌───────────┐ ┌─────────────┐ ┌────┐ ┌──┐ ┌─────────┐
 wh students    do pres plur    have   no   advisors?
```

Transformed rules:

```
       1              #2 + n' + #             3     0       5
┌───────────┐ ┌────────────────────────┐ ┌──────┐ ┌──────────┐
 wh students    #do+pres+plur+n't#         have      advisors?
```

where # indicates that the result must be treated as a unit for further transformations. The morphology routines will combine the result to produce "don't".

7. Other Features of the Paraphraser

The paraphraser is used for a second purpose in addition to paraphrasing. It can generate an English version of the parser's representation as well as paraphrase in the three-part form. This function uses the same procedures and grammar as the three-part paraphraser, but the tree is not split into three separate trees before being flattened.

In CO-OP, generation is used to produce alternative suggestions and corrective responses. A corrective response is used to correct the user's false presuppositions. When an existential presupposition encoded in the question is incorrect, the portion of MQL representing the failed presupposition (this is determined by CO-OP) is passed to the paraphraser, which generates the corrective response. For example, (K) below is a corrective response that could be generated by the paraphraser if (J) were asked:

(J) Which programmers in division 3 work on projects in oceanography?

(K) I don't know of any projects in oceanography.

Alternative suggestions are also used by the CO-OP system when the direct response to the user's question is negative. If an incorrect presupposition is removed from a question, the resulting question may no longer have a negative response.[11] In such cases, CO-OP suggests the wider class question to the user as a possible interest. CO-OP passes the MQL representing this question to the paraphraser, which generates the English for the suggestion. A sequence like (J), (K) above might be followed by the alternative suggestion (L):

(L) But you might be interested in programmers in division 3 that work on any projects.

For both types of responses, the paraphraser generates the response using the paraphrase functions with minor differences. The flattening process for generation differs from that used for paraphrases in that the tree is not divided into subtrees representing given and new information and, therefore, the tree is flattened as a whole. The transformational grammar also applies to the generation process, with the one difference being the point at which the applicability of wh-fronting is tested for (described in Section 6.4). Other than these changes and the use of different leading phrases (e.g., "But you might be interested in ..."), the generation process is the same as the paraphraser process. The generation function is general enough that it could be used for other types of responses in cases when something other than a direct response is needed.

8. Related Research

At the time of the CO-OP paraphraser implementation, two main other paraphrasers had been developed and implemented for data base question-answering systems:

- ▸ PLANES, Waltz et al. 1978;
- ▸ RENDEZVOUS Version 1, Codd 1978.

Both systems used *templates* to form the paraphrases. Templates are canned English phrases (or sentences) containing slots that may be filled with different words to produce a variety of full English phrases.

The PLANES system generates the paraphrase from the formal data base query using templates. The process involves three specific actions. English words are substituted for any abbreviations or code names in the

[11] See Kaplan 1979 for details on determining the most appropriate alternative suggestion.

data base query, using a table look-up. A single appropriate paraphrase template is selected for use based on the query, and the slots in the template are then filled with words and phrases from the query. The major effort in designing this kind of system is in the formation, by hand, of templates suitable for the particular data base and for the types of questions that can be asked. An example of an English question and the PLANES paraphrase for it are shown below in (M):

(M) Q: How many flights did plane 3 make in Jan 73?
P: PLANES searches the MONTHLY FLIGHT and MAINTENANCE SUMMARIES and returns: The value of TOTAL FLIGHTS for plane SERIAL #3 during January 1973.

The RENDEZVOUS system also generates the paraphrase from the formal query using templates, although it is slightly more sophisticated than Waltz's. There are three parts to generation, and two types of templates are used. A header template corresponding to the type of query is chosen first. There are three types of queries in the system (FIND, EXIST, COUNT), of which FIND occurs most frequently. The header for FIND is PRINT THE ... EVERY ..., where the dots must be filled in. The second part to the paraphrase is the target list. It specifies the attributes requested by the user and is supplied by doing a table look-up on the attribute. The third part of the paraphrase is called the body. It is formed by extracting templates from tables, associated with particular items in the query, that specify restrictions on the requested values. An example of a query and the paraphrase generated by RENDEZVOUS is shown in (N) below.

(N) Q: I want to find certain projects. Pipes were sent to them in Feb. 1975.
P: Print the name of every project to which a shipment of a part named pipe was sent during February 1975.

The goals of the RENDEZVOUS generation component are important ones. The generated English must be unambiguous, easy to understand, discriminating, and not misleading (Codd 1978). Instead of developing a general solution to achieve these goals, however, the research seems to be concentrated on particular examples which don't meet these criteria. This results in part from the use of templates. The templates must be constructed beforehand for a particular data base, and great care must be taken to choose phrases that can be easily patched together with a variety of other phrases. Unforeseen interaction between juxtaposed phrases is a problem that frequently arises. Such an approach necessitates looking at particular examples, instead of the general framework.

In both of these systems, the use of templates means that the major effort in developing the system must be done by hand in formatting the English phrases. All questions that will be asked must be anticipated ahead of time, and although the systems can be extended by adding new templates, undesirable interactions between new and old templates must be specifically avoided, and each new required addition does not ease the addition of subsequent templates. Note that this means coverage in a template system is also difficult to specify.

The use of a grammar in the CO-OP paraphraser makes it more flexible than these earlier paraphrasers:

- ▶ less work must be done by hand in formulating the system,
- ▶ interactions between templates are not a problem since the grammar determines how to combine words and phrases in an acceptable way, and
- ▶ the system is capable of handling new questions for which it has not been explicitly prepared, as long as they fall within the syntactic range of the system.

The paraphraser's ability to perform the generation task described in the previous section nicely illustrates its flexibility. Note furthermore that the CO-OP paraphraser specifically addresses the problems of disambiguating relative clause modification in a general way and of generating a paraphrase that differs from the original question on a theoretical basis, issues not addressed by either the PLANES or the RENDEZVOUS paraphraser.

9. Conclusions

The paraphraser described here is a syntactic one. While this work has examined the reasons for different forms of expression, additions must be made in the area of semantics. The substitution of synonyms, phrases, or idioms for portions or all of the question requires an examination of the effect of context on word meaning and of the intentions of the speaker on word or phrase choice. The lack of a rich semantic base and contextual information dictated the syntactic approach used here, but the paraphraser can be extended once a wider range of information becomes available.

When testing the implementation of the CO-OP system and extending its linguistic coverage, the paraphraser proved particularly helpful in debugging incorrect parses. It provided fast, easy-to-recognize notification when an incorrect interpretation had been made. This leads us to believe that the paraphrase would also prove helpful to actual users of the system were CO-OP to interpret a question differently than it was intended. Testing of this facility with a large number of actual users remains a topic for future work.

The CO-OP paraphraser has been designed to be domain-independent, and thus a change of the data base requires no change in the paraphraser. Paraphrasers that use the template form, however, will require such changes. This is because the templates or patterns, which constitute the type of question that can be asked, are necessarily dependent on the domain. Different sets of templates must be used for different data bases.

The CO-OP paraphraser also differs from other systems in that it generates the question using a transformation grammar of questions. It addresses two specific problems involved in generating paraphrases:

1. ambiguity in determining which noun phrases a relative clause modifies;
2. the production of a question that differs from the user's.

These goals have been achieved for questions using relative clauses through the application of a theory of given and new information to the generation process.

Acknowledgments

This work was partially supported by an IBM fellowship and NSF grant MC78-08401. I would like to thank Dr. Aravind K. Joshi and Dr. Bonnie Webber for their invaluable comments on the style and content of this paper.

References

Akmajian, A. and Heny, F. 1975 *An Introduction to the Principles of Transformational Syntax.* Academic Press, New York, New York.

Chafe, W.L. 1977 Givennness, Contrastiveness, Definiteness, Subjects, Topics, and Points of View. In Li, C.N., Ed., *Subject and Topic.* Academic Press, New York, New York.

Codd, E.F. et al. 1978 Rendezvous Version 1: An Experimental English-Language Query Formulation System for Casual Users of Relational Data Bases. IBM Research Report RJ2144, IBM Research Laboratory, San Jose, California.

Cullicover, P.W. 1976. *Syntax.* Academic Press, New York, New York.

Danes, F., Ed. 1974 *Papers on Functional Sentence Perspective.* Academia, Prague.

Firbas, Jan. 1966 On Defining the Theme in Functional Sentence Analysis. In *Travaux Linguistiques de Prague 1.* University of Alabama Press.

Fibras, Jan. 1974 Some Aspects of the Czechoslovak Approach to Problems of Functional Sentence Perspective. *Papers on Functional Sentence Perspective.* Academia, Prague.

Goldman, N. 1975 Conceptual Generation. In Schank, R.C., Ed., *Conceptual Information Processing.* North-Holland Publishing Co., Amsterdam.

Grice, H.P. 1975 Logic and Conversation. In Cole, P. and Morgan, J.L., Ed., *Syntax and Semantics: Speech Acts,* Vol. 3. Academic Press, New York, New York.

Halliday, M.A.K. 1967 Notes on Transitivity and Theme in English. *Journal of Linguisticsx* **3.**

Heidorn, G. 1975 Augmented Phrase Structure Grammar. In *TINLAP-1 Proceedings.*

Joshi, A.K. 1979 Centered Logic: The Role of Entity Centered Sentence Representation in Natural Language Inferencing. In *IJCAI Proceedings.*

Kaplan, S.J. 1979 Cooperative Responses from a Portable Natural Language Data Base Query System. Ph.D. dissertation. University of Pennsylvania, Philadelphia, Pennsylvania.

McDonald, D.D. 1978 Subsequent Reference: Syntactic and Rhetorical Constraints. In *TINLAP-2 Proceedings.*

McKeown, K. 1979 Paraphrasing Using Given and New Information in a Question-answering System. Master's thesis. University of Pennsylvania, Philadelphia, Pennsylvania.

Morgan, J.L. and Green, G.M. 1977 Pragmatics and Reading Comprehension. University of Illinois.

Prince, E. 1979 On the Given/New Distinction, *CLS* **15.**

Simmons, R. and Slocum, J. 1972 Generating English Discourse from Semantic Networks, *Communications of the ACM* **5**(10).

Waltz, D.L. 1978 An English Language Question Answering System for a Large Relational Data Base, *CACM* **21**(7).