Indexing and Dependency Logic for Answering English Questions[1]

This paper describes a computer system which uses a combination of coordinate indexing and structure matching techniques to extract from English questions many criteria which can be used for selecting and recognizing answers. A complete index of all content words in text is first searched to find information-rich statements which may be answers to the question. Each of these statements is then dependency analyzed to determine if the words (or synonyms) which correspond to question words maintain the dependency relations holding in the question. A simple semantic evaluation of structurally acceptable answers follows. A human editor working with the computer system helps to resolve syntactic ambiguities which are otherwise a major stumbling block in question-answering systems.

ROBERT F. SIMMONS, SHELDON KLEIN, and KEREN McCONLOGUE

System Development Corporation
Santa Monica, California

• Introduction

One of the most attractive approaches to language processing on computers is the research on program systems that enable the computer to answer natural English questions. The answers are variously proposed to be fragmented lists of information, selections of sentences and paragraphs from text or, most recently (6), the generation of a new statement combining the question with some fragments of text that appear to answer it.

The appeal of this avenue toward language processing lies in the utter generality of the problem of answering questions. Any statement, no matter how complex, can be transformed into a question. Thus any analysis suitable for all possible questions must cover the breadth of the language.

Yet, despite their breadth, question-answering systems lend themselves readily to fractionation into small packages. For example, Green (2), by prestoring his language input data into a meaningful structure, was able to concentrate entirely on deriving some of the operations that a question implied for the data store. Earlier, Lindsay (7) had specialized on the problem of where to put a given name in a family tree when his system read a statement about family relationships. More recently, Kirsch (9) has developed a system for answering yes or no to a limited set of questions about geometric figures in a two-dimensional space.

The Synthex research project of the authors has attempted to deal broadly with many aspects of question answering. This approach is seen as a road toward developing a general-purpose language processor that will synthesize some of the complex human cognitive processes involved in language usage. The prototype research vehicle, Protosynthex I, accepts ordinary text and English questions as its inputs. It indexes the text and analyzes the questions into the terms of the index. The terms of a question are looked up in the index, and potential answering statements are selected from the text and scored for relevancy to the question. At that point the most challenging task begins — that of evaluating whether or not a statement is an answer to the question.

This paper is designed to show how a question contains many criteria for recognizing its answer and how some of these criteria may be used to find and evaluate potential answering statements. The techniques discussed have been developed as working computer programs for various IBM digital computers and are used as part of Protosynthex I.

• The Nature of Questions

A complete question is one for which it is possible to obtain answers such as "yes," "no," or "to a certain

[1] This research was conducted under ARPA contract SD-97.

extent." Some examples are: "Did John go to the store?" "Finished breakfast yet?" "Do worms eat grass?" In every case, the only information lacking is the knowledge of whether the assertion is true or false. Other English questions are less complete; e.g., "Where did John go?" "What do worms eat?" Later in this paper, a close examination of the incomplete question will show that the question words — "who," "what," "where," etc. — contain a great deal of information which is useful in finding and evaluating potential answers.

The first problem in answering a question is to understand what it is asking about. Considering only the words contained in the question, it can be seen that a vast amount of information is usually present. The words used in a question can be categorized into three large classes: the question words "who," "what," "where," etc.; the function words, which indicate grammatical constructions, such as articles, prepositions, conjunctions, etc.; and finally the content words such as nouns, verbs, adjectives, etc., which carry the bulk of the meaning.

In terms of information theory the selection of content words in a question represents a choice of five or ten words from a population of words numbering somewhere between 75 and 150 thousand. The amount of information contained in each word is roughly equivalent to the inverse of its probability of selection. Thus the content words selected from a list of 100,000 words contain far more information than the function words selected from a list of a few hundred words or the question words selected from a list of less than a dozen. The content words, therefore, are the most significant in discovering possible answers to a question.

In the system described below, the content words are index entries. The index cites the locations of all statements in a text that use these content words and makes it possible to retrieve information-rich statements — not answers necessarily, but data pertinent for answering questions. Some of the function words and all of the question words are also essential cues for evaluating possible answers to questions.

The English question words — "who," "what," "where," "how," etc. — all carry very special meanings and substitute for certain semantic classes of words. "Who" substitutes for a person, "where" for a place, "when" for a time and so on; thus they indicate the *type* of word or phrase required in an answer. In addition, they also signify syntactic classes of words. "Who" and "what," for example, show that the answer should be in a nominal or noun-phrase construction. "Where" and "when" require a verb-modifier construction. In the latter case, the question words also select a small set of prepositions or adverbs such as "in," "on," "at," "near," etc. "How" is answered by an overlapping set of prepositions, though still usually in a verb-modifying construction. These cue constructions include "by means of," "with," "by use of," "by —ing," etc.

By using the information contained in the question words, such an incomplete question as "What do worms eat?" may be transformed into a statement that will help to identify the answer. The first step is to transform the question into an assertion; e.g., "Worms eat what." (Just how this is done will be described later.) The question can now be further transformed into "Worms eat X; X=thing, nominal." "Where do worms eat" would transform into the following:

> "Worms eat X; X=(prep/adv place, and place, nominal)."

If the potential answering statement includes "worms," "eat," and a place word modifying the verb "eat," *and if all these words are in an appropriate set of relationships*, then it is known that an answer is present. (Whether or not the answer is true is yet another difficult question still to be dealt with.)

But just exactly what is an appropriate syntactic arrangement of the question words to allow for a possible answer? Such acceptable answers as the following come readily to mind for the question "What do worms eat?"

> Grass is eaten by worms.
> A worm-gnawed apple. . .
> Worms eat their way through the ground.
> Worms eat grass and bits of vegetation.

Some unacceptable answers follow:

> Birds eat worms.
> Horses with worms eat grain.
> Worms are eaten by birds.

It is to be noticed that no easy prediction can be made as to just where in an answering statement the things that worms eat are to be found. Truly the answers to a "what" are always in nominal constructions, but that fact is not in itself enough to determine that a statement is an answer. What is pertinent is that the terms in the answering assertion bear the same set of interrelationships as they do in the question.

To discover whether the relations are parallel between the question and the potential answer, a detailed syntactic analysis must be made. While the awkwardness of comparing syntactic structures is well known (2, 4) one aspect of the syntactic analysis can be compared easily from question to answer. The dependency relations of the answer must be essentially the same as those of the question.

• Phrase Structure and Dependency Analysis

A phrase structure analysis of a sentence depicts relations among individual words or groups of words. A dependency analysis is concerned only with relations among individual words — relations of dependency or modification.

While a dependency analysis can be derived from a phrase structure analysis of a sentence, the reverse is not true unless additional information is used. Accordingly, a dependency analysis contains less information,

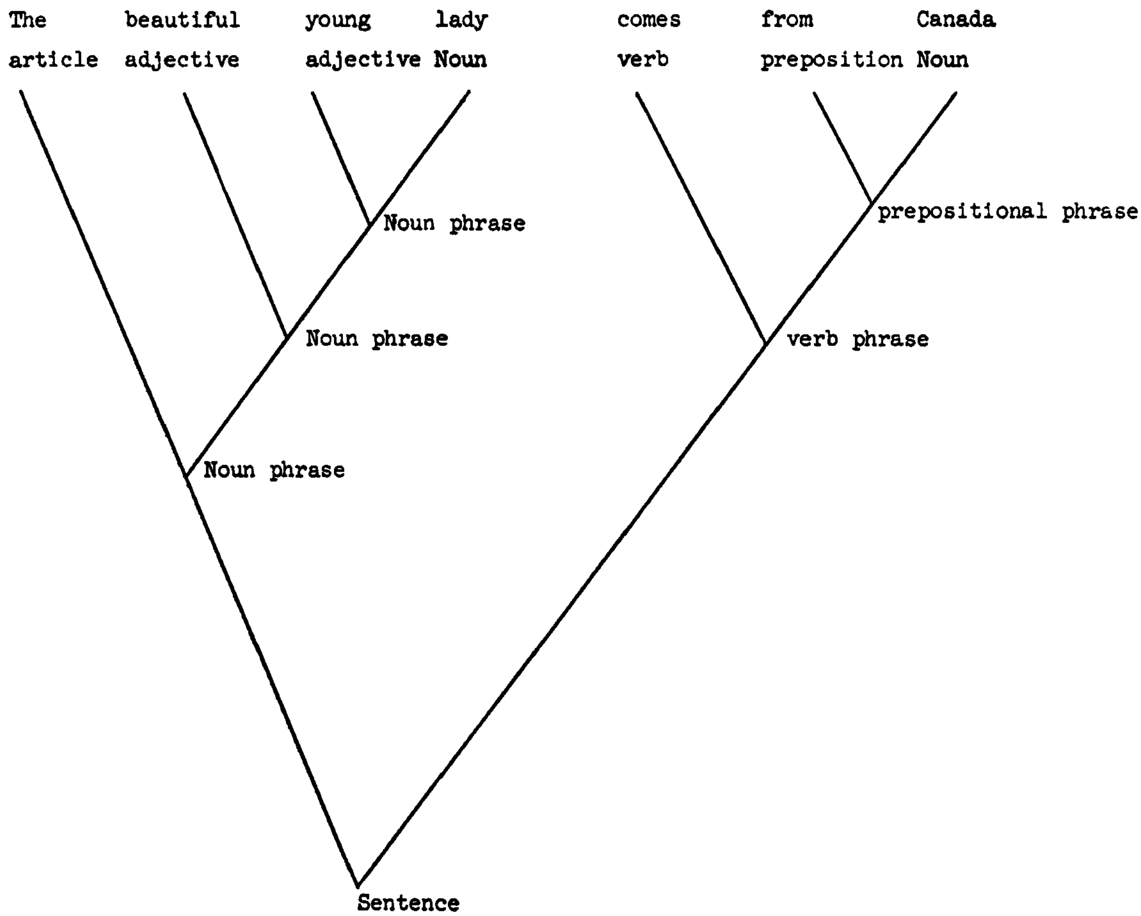

FIG. 1. Phrase Structure Analysis of a Sentence

but that information can be coded in the notation of a single binary relationship; i.e., one word is or is not dependent upon another. Such a feature permits extremely simple computer handling of complex syntactic features.

A phrase structure analysis can be in the form of a tree structure whose nodes are labeled with the names of constructions they represent, as illustrated in Fig. 1. A dependency analysis can also be in the form of a tree structure, as illustrated in Fig. 2.

Note that in Fig. 2, "the" and "beautiful" are directly dependent upon "lady," although they are separated from "lady" by intervening items in the actual sentence. The reason for this can be suggested by a brief examination of the notion of a phrase structure generation grammar (2, 12). The sentences of Fig. 2 can be generated by iterative application of a set of rewrite rules of the following type:

1. sentence = noun phrase + verb phrase
2. noun phrase = article + noun phrase

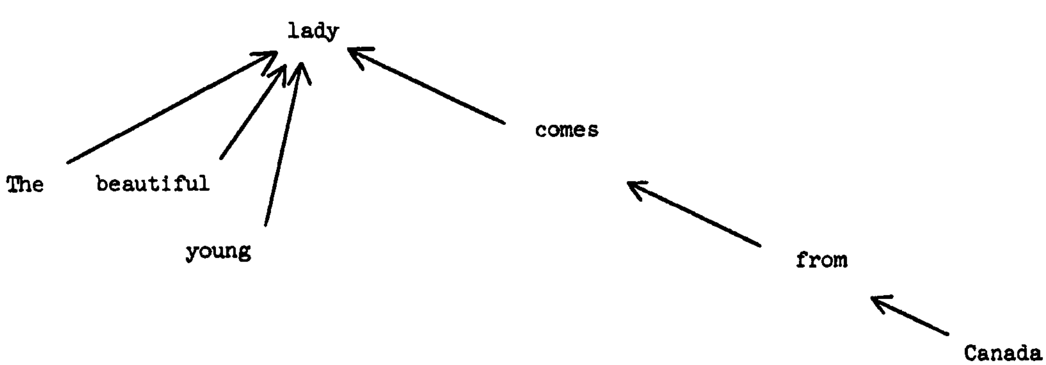

FIG. 2. Dependency Analysis of a Sentence

3. noun phrase = adjective + noun phrase
4. verb phrase = verb phrase + prepositional phrase
5. prepositional phrase = preposition + noun phrase
6. noun phrase = noun
7. verb phrase = verb
8. noun = lady
9. noun = Canada
10. article = the
11. adjective = beautiful
12. adjective = young
13. verb = comes
14. preposition = from

In using these rules, the term on the left of the equal sign is to be rewritten as the term or set of terms on the right. Thus "sentence" is rewritten as "noun phrase + verb phrase." This set of terms is similarly expanded by rewriting each element; thus "noun phrase" is rewritten as "article + noun phrase," etc. Following these substitution rules the example sentence might be produced as follows:

a. sentence
b. noun phrase + verb phrase (by rule 1)
c. article + noun phrase + verb phrase (by rule 2)
d. the + noun phrase + verb phrase (by rule 10)
e. the + adjective + noun phrase + verb phrase (by rule 3)
f. the + beautiful + noun phrase + verb phrase (by rule 11)
g. the + beautiful + adjective + noun phrase + verb phrase (by rule 3)
h. the + beautiful + young + noun phrase + verb phrase (by rule 12)
i. the + beautiful + young + noun + verb phrase (by rule 6)
j. the + beautiful + young + lady + verb phrase (by rule 8)
k. the + beautiful + young + lady + verb phrase + prepositional phrase (by rule 4)
l. the + beautiful + young + lady + verb + prepositional phrase (by rule 7)
m. the + beautiful + young + lady + comes + prepositional phrase (by rule 13)
n. the + beautiful + young + lady + comes + preposition + noun phrase (by rule 5)
o. the + beautiful + young + lady + comes + from + noun phrase (by rule 14)
p. the + beautiful + young + lady + comes + from + noun (by rule 6)
q. the + beautiful + young + lady + comes + from + Canada (by rule 9)

Note that at step *d*, "the" was immediately adjacent to the noun phrase unit, although it later became separated. Similarly, at step *f*, "beautiful" was also adjacent to that noun phrase. It is immediate contiguity of elements *at some stage in the generation process* that helps to determine the dependency between elements which are physically separated in the final sentence.

This discussion of phrase structure and dependency is extremely simplified and ignores criteria for the order of application of the phrase structure rules. A detailed discussion of the relationship between the two, including an algorithm for deriving a dependency analysis from a phrase structure analysis, can be found in the work of Klein (6).

Some rules for deriving a dependency analysis from rather general syntactic criteria (6, 7) are shown below. (For a different type of dependency analysis see the work of Hays [5].)

1. The head of the main verb phrase of a sentence or clause is dependent upon the head of its subject.
2. The head of a direct object phrase is dependent upon the head of the governing verb phrase.
3. Objects of prepositions are dependent upon those prepositions.
4. Prepositions are dependent upon the heads of the phrases they modify. Prepositions in the predicate of a sentence are dependent upon the head of a verb phrase and also upon the head of an intervening noun phrase if one is present.
5. Determiners and adjectives are dependent upon the head of the construction in which they appear.
6. Adverbs are dependent upon the head of the verb phrase in which they appear.
7. Two-way dependency exists between the head of a phrase and any form of the verb "to be" or the preposition "of." This rule holds for the heads of both phrases linked to these forms.
8. Two-way dependency within or across sentences also exists between tokens of the same noun and between a pronoun and its referent.
9. Dependencies within a passive sentence are treated as if the sentence were an active construction.
10. The head of the subject is dependent upon itself or upon a like token in a preceding sentence.

• Answering Questions

So far we have discussed the kinds of information contained in a question that can be used to identify an answer. The content, question, and function words of the question have all been shown to offer important cues for identifying an answer. In the following two sections we shall describe how the information-rich statements which may contain answers are found, and the techniques for using the dependency analysis to select those statements which are most probably valid answers.

SELECTING INFORMATION-RICH TEXT IN RESPONSE TO A QUESTION

A first requirement on any question-answering system is an organized storage of data. In an earlier paper on maximum-depth indexing (11), Simmons and McConlogue described the Indexer, a system for impressing on running text an organization suitable for extracting potential answers from that text. The Indexer constructs a complete index of all the content words in the text and cross-references such words as "elect," "elections," "electing," "elects," etc. For the Protosynthex I system, a complete index of the *Golden Book Encyclopedia* was made, and this text serves as the basis for the question-answering system.

Questions are inserted into this system by punched cards (teletype or flexowriter may be used if desired). The question may be any combination of English words not to exceed 20 or 25 (depending on the length of each word), followed by a question marker. The first analysis sorts the words into content, function, and question words. Next, the content words extracted are used to find references in the index to every occurrence in text of a sentence using these words. The references are in the form of VAPS numbers (Volume, Article or chapter, Paragraph, Sentence).

Thus for a hypothetical (and very vague) question such as "What farmers were elected?" the content words "farmers" and "elected" would be selected and the VAPS numbers recovered would represent all sentences that contain either or both words. Such a list is illustrated in Table 1. A first step in discovering potential answers is to search for identical VAPS numbers for each of the words. In Table 1, the VAPS number 1-17-3-1 is common to both words. If no such *sentence* had been found we would have been interested in any *paragraph* in which the two words were found. Other things being equal, those sentences, paragraphs, or chapters which contain intersections of the content words are the most likely candidates for answers.

However, other things are not equal in several ways. First, for a one-content-word question such as "What is a farmer?" no intersection is possible. For this case, the paragraph or article containing the most references to that content word is selected.

A second case arises from the fact that not all words are equally important for finding an answer. Consider the words "type" or "kind" and the word "zebra." The first words are hardly more than markers while the latter points precisely to a single meaning. If the question is asked, "What kind of farmer was elected?" the word "kind" changes the meaning to a relatively small extent from the original question. An interesection of all three content words will hardly bring back any more satisfactory answer than will the two previously considered.

In general, the more frequent a content word the vaguer its meaning; the more infrequent the word the more precise its meaning.[2] The inverse of a word's frequency in a large corpus of text is thus approximately proportional to its importance as an indexing term for finding information-rich statements. Consequently, as a word is looked up in the index, its number of occurrences in the large sample of text is used as its relative frequency count. The inverse of this count is used to derive an information score for the question and possible answering statements. For example:

"What kind of farmer was elected?"

Frequency:	30	10	5
Inverse:	1/30+1/10+1/5=1/3=.333		

The sum of the inverse frequencies for the question is called Qmax. A similar procedure for each answer results in a sum called Amax. The ratio Amax/Qmax is a measure of how closely the information content of words in the answer matches that of the question. Although the statistic is admittedly a first approximation, it has the desired property of weighting most heavily those content words which carry the most information. (In the above example, Qmax would be .300 without the word "kind"; thus an answering statement that contains "farmer" and "elected" would give an information-rich ratio of .300/.333 =.90.)

A third case in which the simple intersection is not sufficient occurs when some or all of the content words in the question have no correspondents in the index. For this case a dictionary of synonyms is gradually being developed. Entries for the synonym list are derived empirically by discovering what words in failed questions would have brought back appropriate text.

After the intersecting and scoring phase has been completed, the surviving VAPS numbers are used to find the actual sentences, paragraphs, or articles which have been estimated to be relevant to the question. As was described in the article on the Indexer (11), the encyclopedia which serves as a text base has been organized on magnetic tape in terms of volumes, articles, and paragraphs. The VAPS numbers give addresses to the location of pertinent text, and it is a simple matter to spin the tape reel once to retrieve all the pertinent statements. The retrieved text along with the questions are then input to the grammar machine for further analysis.

TABLE 1. VAPS Recovery for "Farmer" and "Elected"

Entry	Volume	Article	Paragraph	Sentence
Farmer	1	1	5	4
	1	1	5	7
	1	17	3	1
	1	17	3	2
Election	1	1	5	2
	1	17	3	1
	1	35	4	1
	1	35	4	2
	1	36	2	2

RECOGNIZING ANSWERS

To discover just what elements of a question are relatively invariant in all answering constructions, it is necessary to make a dependency analysis of both the question and the proposed answers. Fig. 3 shows such analyses for the examples cited as potential answers to "What do worms eat?" In analyzing this question it is apparent that "worms" is the main noun phrase or subject, "eat" is the verb modifying "worm," and "what" is the object modifying "eat." The word "do" belongs to a category of words used to signify the question transform and drops

[2] See, for example, Cherry (1).

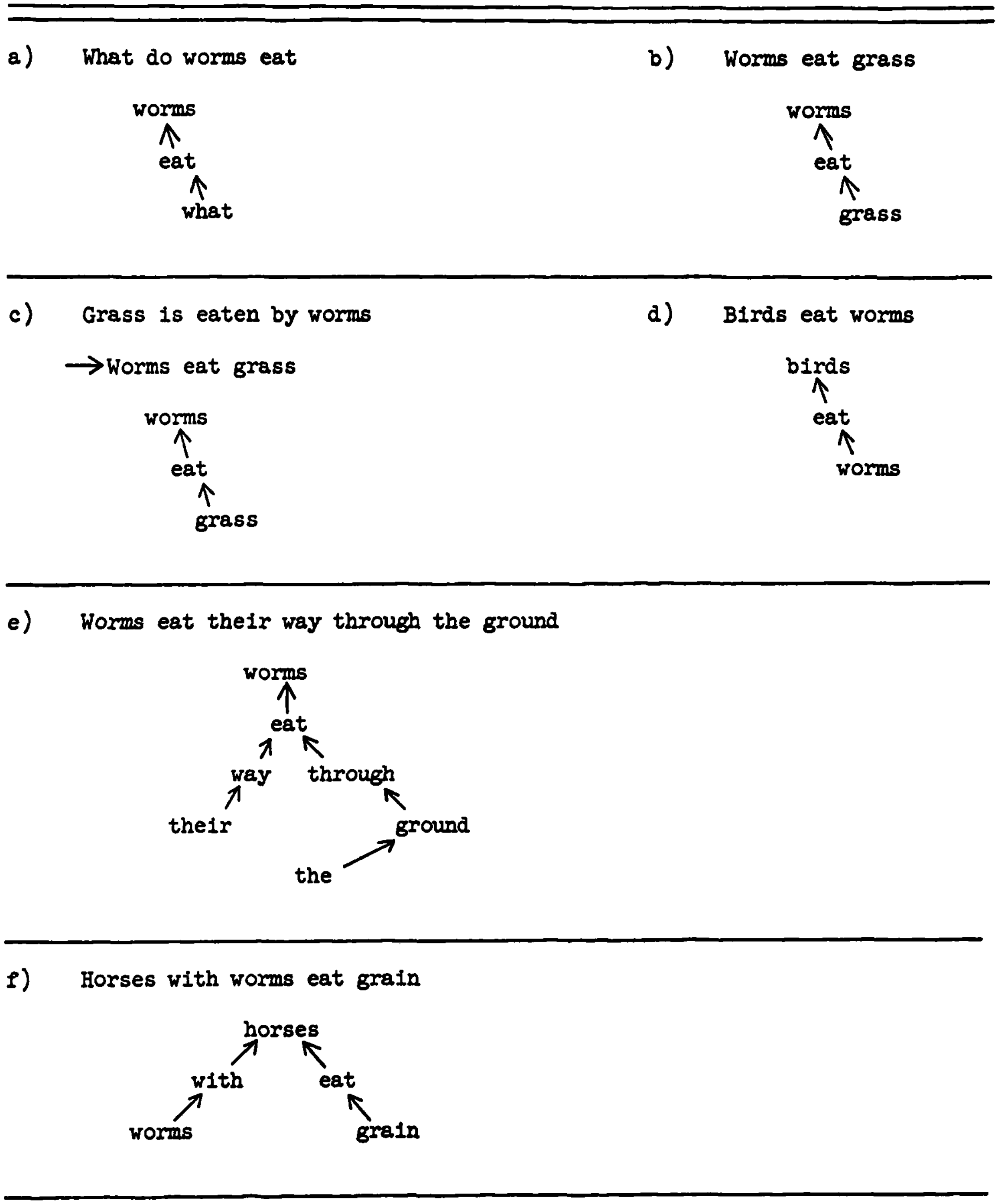

FIG. 3. Dependency Structures of a Question and Some Potential Answers

from consideration. In Fig. 3*a* the tree structure of this analysis is shown.

Each of the statements is similarly analyzed for dependencies. Every passive statement — e.g., "Worms are eaten by birds" — is transformed to an active one — "Birds eat worms." Fig. 3 *b* shows the analysis of "Worms eat grass." "Eat" is dependent on "worms," and "grass" is dependent on "eat." The dependency relations of the answer are precisely those of the question with "grass" filling the position of "what." The statement shown in Fig. 3*c*, "Grass is eaten by worms," is the passive equivalent of "Worms eat grass." After being transformed to an active construction, its dependencies are identical with those of the question.

In contrast, "Birds eat worms" is shown in Fig. 3*d*. Here "eat" is dependent on "birds," and "worms" is dependent on "eat." Since none of these dependencies match those of the question, the answer is rejected.

More difficult situations may be examined in Figs. 3*e* and 3*f*. For the proposed answer "Worms eat their way through the ground," we discover first that "eat" is dependent on "worms" as the question demands. "Way" and "ground" are both nominals and belong to the semantic class demanded by "what." For "way" the dependency match is again perfect and the sentence is accepted. In 3*f*, "Horses with worms eat grain" is readily seen to fail the requirement that "eat" be dependent on "worms," and may be rejected. Two senses of "with" are possible in the sentence: "Horses *and* worms eat grain," or "Horses *containing* worms eat grain." At present the question evaluation logic is not sensitive to the difference. It might also be argued that if horses with worms inside them eat grain, the worms also eat grain. This inference is also more subtle than can be handled by the question evaluation logic.

A more complex example is offered by the question and some proposed answers shown in Fig. 4. For the question "What mountain range was named after the Appalachee tribe?" the potential answer is in two sentences. "The explorer DeSoto named the Appalachians. He named them after the Appalachee Indians."

The dependency structures of the question and of its proposed answer may best be compared in a matrix as shown in Fig. 5. A "1" in a square of the matrix shows that in the question, the word in the row was dependent on the word in the column. An asterisk with the "1" shows that a word in the proposed answer had the same dependency relation as in the question. It will be noticed

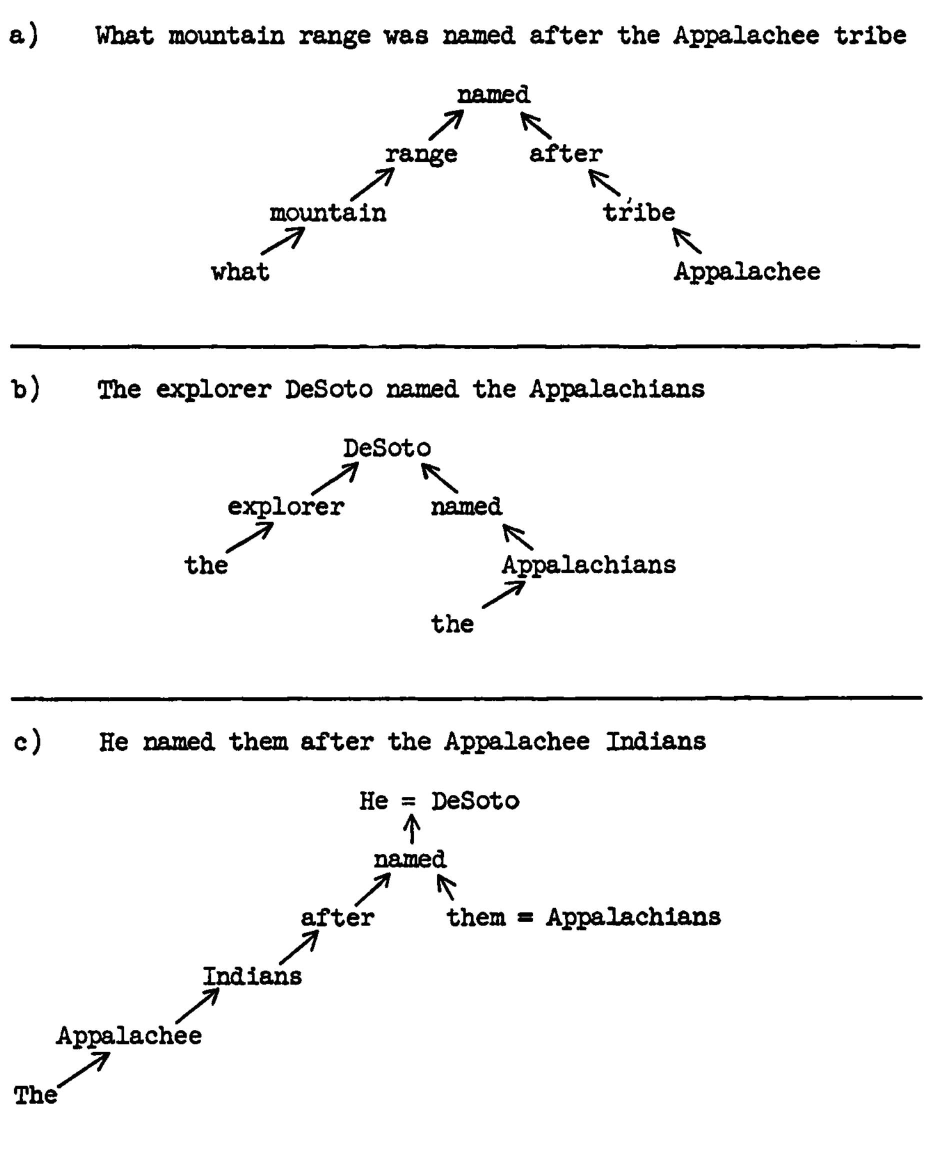

FIG. 4. Dependency Analysis of Complex Structures

first that the question was transformed from passive to active so that it has no subject: "(They) named what mountain range after the Appalachee tribe?" As a consequence, the phrases, "what mountain range" and "after the Appalachee tribe," are both dependent on the verb "named."

The next thing to be noticed in the matrix is that certain words are dependent on others even though they are not adjacent in the dependency tree. This results from the fact that under many circumstances dependency is *transitive*. By transitive we mean that if word "A" is dependent on "B" and "B" is dependent on "C," then "A" is dependent on "C." In the present example this means that where "mountain" is dependent on "range" and "range" is dependent on "named," "mountain" is dependent on "named." Similarly, "what" is dependent on "mountain," "range," and "named."

Transitive dependency appears to hold within nominal and verbal constructions but not across the head of the construction if it is a verb, a preposition, or a subordinate conjunction.[8] Studies of transitive dependency in the context of computer generation of coherent discourse are discussed in earlier papers (6, 7). For question answering, the transitivity of dependence is a primary generalizing feature which allows the recognition of varied forms of English statements as related to the question.

Returning to the matrix of Fig. 5, the asterisks show that the proposed answer held the following dependencies:

"named" *on* (any word)
"after" *on* "named"
(what) *on* "named"
"Appalachee" *on* "after"

A perfect match of answer to question would have given all of the 11 dependencies. Four matches actually occurred. The ratio 4/11 = .36 offers a basis for scoring the sentence. However, the direct scoring value of this statistic for determining correct answers is not expected to be very strong, and a fair amount of effort will be

[8] For some purposes we allow the forms of the verb "to be" and the preposition "of" to be transitive.

	named	range	mountain	what	after	tribe	Appalachee	(any word)
named								1*
range	1							
mountain	1	1						
what	1*	1	1					
after	1*							
tribe					1			
Appalachee					1*	1		

Note: 1 = dependency of question
* = dependency of answer

FIG. 5. Matrix Comparison of Question and Answer Dependencies

required to work out the complex probabilities involved in the combined match of words and relations.

But we have not yet exhausted the information contained in the question. In the proposed answer we discover that "Appalachee" is dependent on "Indians." In the question, "Appalachee" was dependent on "tribe." The question, "Indian = tribe?" may be generated. Similarly "Appalachians" in the answer corresponding to the "what" position of the question can be used to generate "Appalachian = mountain range?" These questions can be processed against the synonym dictionary or against the original text. If the words are not found to be synonymous, the text can be searched to discover if there are statements in which "Indian" is dependent on "tribe," and in which "Appalachian" is dependent on "mountain" and on "range." If these relations are discovered to hold in the larger text, the percentage match is increased, frequently to a rather large extent. The probability that the original statement was an answer would also increase significantly.

A system which would continually generate smaller and smaller questions in its attempt to answer the original query is conceptually very attractive. One can easily visualize such a system being given a large batch of questions — say twenty — to solve. Each of these questions might result immediately in partial answers plus an average of three to five additional questions. Answering these derived questions might generate some lesser number of questions until finally the system had a certain set for which it could find no answers. At this point the computer could make a request for input pertaining to its unanswered questions and so add to its information store.

From a practical point of view, however, such a system might prove extremely expensive. Each question might require several iterations through a large store of information and some might lead to endless recursions.

As an easy alternate, the system could generate such derived questions for the human operator to answer. His answers would enrich the computer storage in the same manner that reading new text would. Thus, by experience, the computer would continually improve its ability to recognize relationships between words.

This approach was chosen for Protosynthex I. The system examines the matrix which compares question-and-answering text as in Fig. 5. It then asks, "Is *Appalachians* dependent on *mountain* and *range?*" and "Is *Appalachee* dependent on *tribe?*" These questions are then answered by a human operator.

A slightly differing form of output has also been contemplated. Given the capability of generating the subsidiary questions above, it is easily possible to print out the following statement:

"If *Appalachians* are dependent on *mountain* and *range*, and *Appalachee* is dependent on *tribe*, then the answer is *Appalachians*." Italic words are those which the computer has found to be critical in answering the question. This kind of output has the attractive (though trivial) feature of insuring that the computer system will always give a logically correct answer.

• Discussion

It was mentioned earlier that this question-answering logic is part of the prototype language processor, Protosynthex I. The flow of operations in this machine includes indexing text, finding information-rich statements in response to a question, analyzing both question and potential answers for dependencies, evaluation through dependency logic of the set of potential answers and finally the beginnings of an analysis of similarity in meaning. There is one striking weakness in the machine: No existing grammatical analysis system can make a completely automatic *unambiguous* parsing of text.

Most syntactic analysis systems have been developed in the context of mechanical translation problems. One of the most sophisticated of these is the predictive analysis system of Oettinger and Kuno (9). Using only syntactic word classes, it discovers dozens of possible interpretations for many, if not most, English sentences. The protosynthex grammar machine is also plagued by ambiguous interpretations. In our case, one interpretation is selected to avoid the rapid multiplication of tree structures. If an incorrect interpretation has been selected, the dependency structure that is derived will usually be wrong and the question-answering logic will fail. At this point a human editor must and does enter the system to correct any errors in the dependency analysis.

At the present stage of development in language processing, ambiguous interpretations can be avoided only by human intervention. When a human parses a sentence, he does so in full knowledge of the meaning of the words which comprise it. The only knowledge the machine can bring to bear is in terms of its word classes and its rules for their combination into phrases, clauses, and sentences. The human's knowledge of meaning may be conceptualized as a much finer set of word classes and combination rules than any computer system has yet been given.

The solution to this problem of ambiguity may lie in the continuous development of finer semantic classifications of words by humans interacting with computer systems. It is our belief that this is one of the more profitable courses to follow, and research efforts with Protosynthex I are directed toward this goal and toward a continually improved understanding of how meaning is encoded into English words and sentences.

References

1. Cherry, C. 1957. *On Human Communication.* New York: Wiley & Sons.
2. Chomsky, N. 1957. *Syntactic Structures.* 'S-Gravenhage: Mouton and Co.
3. Green, B. F., Jr., Wolf, A. K., Chomsky, C., and Laughery, K. 1961. Baseball: An Automatic Question Answerer. *Proc. Western Joint Computer Conf.*, **19:** 219–224.
4. Harris, Z. S. 1957. Co-occurrence and Transformation in Linguistic Structure. *Language,* **33** (3): 283–340.
5. Hays, D. G. 1960. Studies in Machine Translation — 10: Russian Sentence-Structure Determination, RM-2538. The RAND Corporation.
6. Klein, S. 1963. Automatic Decoding of Written English. Unpublished doctoral dissertation, University of California, Berkeley.
7. Klein, S., and Simmons, R. F. 1963. Syntactic Dependence and the Computer Generation of Coherent Discourse. *Mechanical Translation,* **7** (2): 50–61. (Also available as SDC document TM–758/000/00.)
8. Lindsay, R. K. 1960. The Reading Machine Problem. Unpublished doctoral dissertation, Carnegie Institute of Technology.
9. Oettinger, A. G. 1963. Mathematical Linguistics and Automatic Translation. Report to the National Science Foundation, No. NSF–8, Cambridge, Mass. (Jan.).
10. Rankin, B. K. 1961. A Programmable Grammar for a Fragment of English for Use in an Information Retrieval System. National Bureau of Standards Report No. 7352. (June).
11. Simmons, R. F., and McConlogue, K. 1963. Maximum-depth Indexing for Computer Retrieval of English Language Data. *Am. Doc.,* **14** (1): 68–73. (Also available as SDC document SP–775.)
12. Yngve, V. H. 1960. A Model and an Hypothesis for Language Structure. *Proc. American Philosophical Society,* **104** (5): 444–466.