BOLT BERANEK AND NEWMAN INC

CONSULTING · DEVELOPMENT · RESEARCH

AFCRL-66-620

AD639323

PROBLEMS IN NATURAL LANGUAGE COMMUNICATION WITH COMPUTERS

Daniel G. Bobrow

Bolt Beranek and Newman Inc.
50 Moulton Street
Cambridge, Massachusetts 02138

Contract No. AF19(628)-5065

Project No. 8668

Scientific Report No. 5

August, 1966

(The work reported was supported by the Advanced Research Projects Agency, ARPA Order No. 627, Amendment No. 2, dated 9 March 1965.)

Prepared for:

AIR FORCE CAMBRIDGE RESEARCH LABORATORIES
OFFICE OF AEROSPACE RESEARCH
UNITED STATES AIR FORCE
BEDFORD, MASSACHUSETTS

Distribution of this document is unlimited.

ABSTRACT

This paper gives an overview of the problems involved in the construction of a computer-based question-answering system designed to interact with the user in English. The system is viewed as containing five distinct parts — a parser, a semantic interpreter, an information storer, an information retriever, and an English output generator. There is a need for extensive interaction among these subsystems, and between the subsystems and the user. Examples are given of the type of processing done by each subsystem, and the nature of the possible interactions. The syntactic analysis described is based on a Chomsky type of transformational grammar. The semantic store is characterized by a form of the predicate calculus, with additional algorithms for computation, and structures designed for fast access to relevant data.

SECTION I

INTRODUCTION

With the advent of time-shared computers, an interactive computing facility has been placed at the command of large numbers of people. However, the computer remains inaccessible to many because they cannot speak a language that the computer understands. Development of programs which "understand" natural language (in the sense described below) will allow interaction between man and machine to take place, for example, in English.

There are a number of projects throughout the country which are attempting to develop computer systems which will accept English sentences typed into the computer, and respond to the user in English. This article is based on work in progress at Bolt Beranek and Newman Inc. on one such system, called SENSE (SEmantic Network Storage Experiment). This system is not yet completely implemented, and thus some of what follows should be considered in the nature of speculation. We present here only an overview of some of the problems (mostly unsolved) involved in development of natural language communication with computers. We use simple examples to illustrate these problems, and go into no detail on solutions.

SECTION II

NATURAL COMMUNICATION

Before proceeding further, let us warn the reader that natural language is not always the most natural medium for communication. A physicist describing the motion of objects in a gravitational field uses a differential equation because it is more natural, and, more important, more precise. An architect feels a picture is worth a thousand words. Both use their own languages (mathematics and graphics) to interact with computers. Even in cases where English would do, when the same wordy message must be transmitted again and again, or very rapidly, people use codes or push buttons to "converse" with computers.

When, then, is natural language "natural?" — in cases where no code or jargon has been generated, where messages are seldom repeated, or where the ideas to be transmitted are not really precisely defined. An important aspect of the communication between people is that a listener, by asking the right questions of a speaker, forces him to define more carefully the relationships he is describing. In our efforts to build computer systems to understand natural language, we cannot ignore this important type of interaction. A good natural-language system must be active, not passive, and must work with the user to achieve an understanding.

We have used the word "understand" in connection with a computer system. Let us provide an operational definition of this concept. We cannot talk about a computer "understanding" a sentence in vacuo. The computer understands only in the context of a body of information and procedures which it contains. All of the systems under construction contain a fairly complex store of information and set of procedures. The critical test of understanding is, can the system answer a question phrased in English based on its stored information? The "depth of understanding," or the problem-solving ability of the program, can be measured by how "obvious" the answer is from the data base; retrieval of a prestored fact is simpler than a deduction or computation based on, or implicit in, the data base.

There are limitations to the English that will be accepted by natural-language interactive systems (including people). SENSE is no exception. These limitations fall broadly into two classes — syntactic and semantic. Syntactic limitations deal with the forms of statements or questions the system will accept. For example, the system may recognize only "active sentences" such as "John read the book," and not accept passive forms such as "the book was read by John." Semantic limitations refer to the kind of things you can talk about. The STUDENT program,[3] for example, could answer only questions framed in the context of algebra story problems; for example:

Mary was twice as old as Ann was when Mary was twice as old as Ann is now. If Mary is 24, how old is Ann?

Within this very limited context STUDENT demonstrated a remarkable understanding (i.e., question-answering ability), but was incapable of understanding things outside this semantic domain. Similarly, the question-answering system currently under development at Technical Operations, Inc., Burlington, Massachusetts (personal communication), can answer only questions relevant to its data base, which is a coded form of the airline guide. Questions outside this context are rejected as not understandable.

These limitations effect the usefulness of a natural language system. Some other critical questions, however, are: how comfortable is a person confined within these limits in communicating with the system; and how easily can the system be extended to include new syntactic structures and semantic relations? We will discuss these questions after we describe the structure of the semantic store and the form of grammar of the language.

The flow diagram in Figure 1 illustrates the general structure of the "SENSE" question-answering system.

The solid lines indicate flow within the system when the system can process an input without further help from the user.

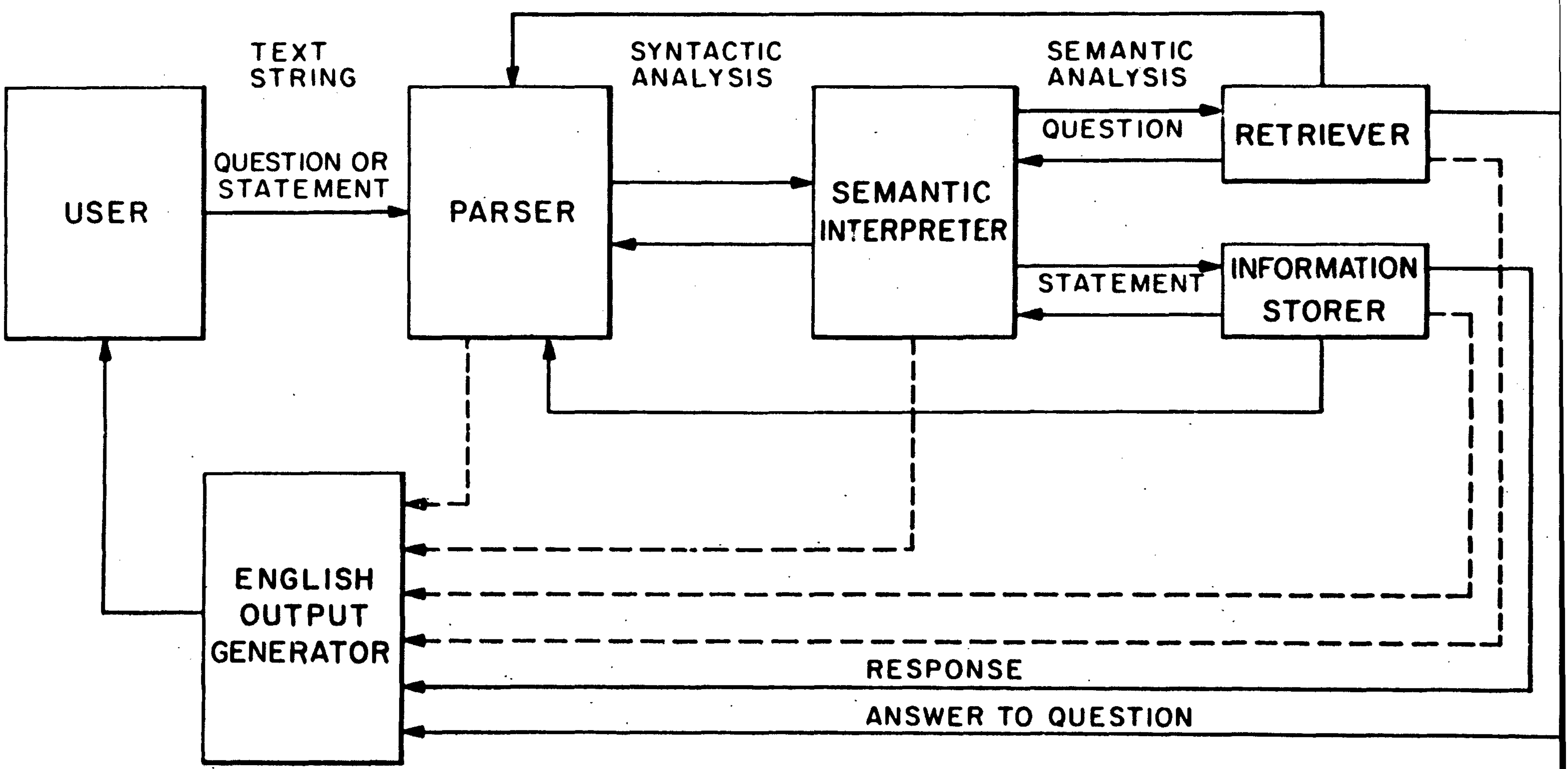

FIG. 1 FLOW DIAGRAM OF THE SENSE QUESTION ANWERING SYSTEM

The dotted lines represent queries back to the user for help. For example, if the user says, "John likes flying planes," the system might inquire whether "flying" should be considered a verb or an adjective in this sentence — thus eliminating this syntactic ambiguity before further processing. Similarly, the semantic ambiguity in "Parker is in the pen" can be eliminated by asking the user if he is referencing a fountain pen or pig pen (unless, of course, Parker is a convict). Finally, the retriever might need additional information to determine the answer to a question such as "how long does AA-57 take to get to Los Angeles?" It could ask the user which starting point of the flight, e.g., New York or Chicago, does he wish to consider. All of these queries are directed to the user through a program which converts the computer query into English. No one has yet built a program which transforms a semantic representation of an "idea" or relationship into English output. At present, we compromise by substituting into prescribed forms. At some level this may be all we can really do.

SECTION III

THE STRUCTURE OF THE SEMANTIC STORE

The basic components of the semantic store[2] are five sets — objects, functions, relations, propositions, and semantic deductive rules. The semantic interpretation of an English statement is an assertion that a (set of) relation(s) holds among some specified objects. A proposition is an instance of a relation with particular objects filled in for the arguments. A proposition can be thought of as a "belief" of the system. Suppose we want to add the English statement "John flies the plane" to the semantic store. This would be represented as an instance of the relation *fly* with *John* as the agent argument of the relation, and *plane* as the object. We italicize *fly*, *John*, and *plane* to emphasize the fact that they are not words, but names for relations and objects in the data base.

In addition to these "simple names" for objects, there are "functional names." A function in this system is a mapping from n-tuples of objects into an object. An example of a function is one represented by "*pilot of* ____," and *pilot of the plane* is a functional name which, in the context of the first statement, is another name for the object denoted by *John*. An important (and difficult) part of the question-answering process is determining a simple name (or names, since it may not be unique) which refers to an object denoted by a functional name.

SECTION IV

THE QUESTION-ANSWERING PROCESS

There are two forms of questions that can be asked of the system. The first, which we call a closed question, asks if a given proposition is consistent with the data base. The answer to this type of question may be yes, no, probably, probably not, or I don't know. The answer yes occurs if the system can find this proposition in its data store — or if it can add it to its store by use of its semantic deductive rules. There are three basic types of semantic deductive rules — paraphrase rules, relational deductive rules, and the metarules of the deductive system itself.

A paraphrase rule gives a paraphrase, or first level implication of the use of a word in a sentence, and thus the use of an object or relation in the semantic network. It contains the definition of the words denoting objects and relations in the system. As an example, the following is a transliteration of part of what we store about the relation fly (with the meaning, a pilot flies a plane) in the SENSE system:

1. (Context Information). Fly may be used in a context where there is an agent, call it S, and an object, call it D. S is a person, probably a pilot and D is probably an aircraft.

2. (Paraphrase Information). "S <u>flies</u> D" implies (can be paraphrased as) "S moves in D through the air."

Thus, with this information and the earlier statement, the system can deduce the answer "<u>yes</u>" to the question "does John move?" Also, with this information, this system will answer "probably" to the question "is John a pilot?", using the context information provided. It will also use the context information and the fact that an airplane is not a person to answer "no" to the question "does an airplane fly John?"

The paraphrase rules usually deal with only one relation. The relational deductive rules provide a way of using several propositions to obtain a single proposition. A typical example of a relational deductive rule is the one which defines the transitivity of the relation <u>in</u>, meaning <u>inside of</u>. It states that "A is <u>in</u> B <u>and</u> B is <u>in</u> C <u>implies</u> A is <u>in</u> C" where A, B, and C can be any objects.

Implicit in our use of these two types of rules have been the metarules of the deductive system: the rule for substitution and the rule for detachment. These can be summarized for our system by the following:

1. If there is a rule R_1 <u>implies</u> R_2, with variables, then P_1 <u>implies</u> P_2 is a true statement, provided P_1 and P_2 are derived by substituting any consistent set of objects

for variables in R_1 and R_2.

2. If P_1 is a proposition in the semantic store (i.e., a belief of the system) and P_1 implies P_2 is in the system or can be put in the system by substitution (rule 1), then P_2 can be added to the semantic store. These two rules were both used in the example cited above.

A critical problem in answering questions with such a deductive system is identifying the relevant propositions and relations to make the deductions pertaining to the question. An exhaustive search through a large data base would be impractical, to say the least. For limited data bases, well constructed indexes help cut down the search. For general semantic stores, the problem is more difficult, and no really good solution has yet been found. The search problem is not unique to semantic deductive systems, but is ubiquitous throug most problems in artificial intelligence.[8] In the SENSE system, we will have links to propositions from both a token of the relation and from some of the objects used as arguments in the proposition. The choice of which objects to link to the proposition is dependent on the context, and the relative importance of different objects and relations. This measure of importance represents a "point of view" for the question-answering system. With a point of view, answering some questions will be easier than answering others — with none, all questions will be equally difficult.

The problem of efficient search comes up again in answering "open questions," ones which ask for the name of an object which satisfies a given relationship. An example of an open question is "what city does Flight AA-57 go to?" Of course, the answer may not be unique, and we would expect a question-answering system to give a list of names of all the objects which satisfy the specified relationship. Thus if AA-57 starts from Boston, and stops in Chicago on the way to Los Angeles, the question-answering system should answer "AA-57 goes to both Chicago and Los Angeles."

An alternative form of an open question asks for a simple name of objects denoted by a given functional name. For example, if "destination of _____" were a functional form in the system, then "what is the destination of Flight AA-57?" is an alternative form of the open question stated earlier. Associated with every function in the system must be a program that can compute the value(s) of the function given its arguments. There is an obvious duality between functions and relations. A question can be interpreted as a request for a computation of the value of a function, or a request for a search for objects which satisfy a specified relation. This will depend on whether there is a program available in the system to compute the value of the function, and if it exists, its efficiency relative to a search for the answer.

SECTION V

SYNTACTIC ANALYSIS

Many programs have been developed to perform syntactic analysis of English on the computer.[4] The question we will address here is how such an analysis can be used in the question-answering systems. The first problem which can be attacked with the aid of a syntactic analysis is the determination of the arguments of a relation. From the parsing of "the dog bit the man," we can determine that the agent of the relation denoted by bit is "the dog," and the object "the man." Thus, we can determine the difference between this sentence and "the man bit the dog." The subject of the sentence is the agent argument of the relation in each case.

The problem becomes more difficult if we consider the sentence "the man was bitten by the dog." Although this is a paraphrase of the first sentence, the subject is not the agent of the relation bit. In line with the transformational theory of grammar* as developed by Chomsky,[5] and others, we consider the obvious parsing of this passive sentence only a "surface structure." According to this syntactic theory, the surface structure of a sentence is derived from a deep

---

* Although we are using a transformational theory of grammar, our semantic analysis is very different than that proposed by Foder and Katz[6] for obtaining semantic readings from a transformational analysis of a sentence.

structure by a series of transformations. The passive and and the active forms of a sentence have the same deep structure, and differ on the surface because the optional, and meaning preserving, passive transformation was applied to yield the former. It is this deep structure, then, which allows us to determine the relationship of arguments to a relation in a sentence.

Transformational syntactic analysis helps in resolving another problem. Consider the sentence "the man who wrote the program debugged it." The man referred to in this sentence is an argument in two propositions "man wrote program" and "man debugged program." The deep structure of this sentence shows both embedded sentences in this explicit form, and shows the relationship between the objects named in each. Transformations carry this deep structure to the surface structure in which there is a relative clause, and a pronoun substituted for "program." Thus, the deep structure analysis of this sentence provides the semantic analysis portion of a question-answering system with a form in which each relation in the sentence is explicit.

SECTION VI

CONCLUSION

We have been discussing the problems involved in building a computer-based question-answering system which can converse with a user in natural language. Work has really just begun on these problems. There is not yet even a reasonably complete transformational grammar of English. Even for the sub-grammars that exist, there are no very efficient parsers. Parsing programs tend to give many alternative syntactic analyses, and we are only starting to learn how to utilize information in our semantic store to disambiguate sentences. Some semantic deductive systems have been built*, but none has yet been able to work in a large general data base. The feedback paths among question answerer, semantic analyzer, parser and the user have not really been exploited.

This paper has given a superficial overview of the difficulties facing the designers of computer systems which can communicate in English. Progress in this work will make the computer more accessible to more people for more problems, and lead to a deeper understanding of natural language and the communication process.

---

* The most interesting of these are described in references 3, 7, 9, 10 and 12. Simmons[11] gives a complete descriptive survey of question-answering systems built before 1965.

BIBLIOGRAPHY

1. Black, F., "A Deductive Question-Answering System," Ph.D. Thesis, Division of Engineering and Applied Physics, Harvard University, Cambridge, Mass., June 1964.

2. Bobrow, D. G., "Natural Language Input for a Computer Problem Solving System," Ph.D. Thesis, Mathematic Dept., M.I.T., 1964.

3. Bobrow, D. G., "A Question-Answering System for High School Algebra Word Problems," AFIPS Conference Proceedings, Vol. 26, FJCC, 1964, pp. 591-614, Spartan Press, Baltimore, Maryland, 1964.

4. Bobrow, D. G., "Syntactic Analysis of English by Computer -- A Survey" AFIPS Conference Proceedings, Vol. 24, FJCC, 1963, pp. 365-387, Spartan Press, Baltimore, Maryland, 1963.

5. Chomsky, N., Aspects of the Theory of Syntax, M.I.T. Press, Cambridge, Mass., 1965.

6. Fodor, J. A., and Katz, J. J., "The Structure of a Semantic Theory" in Foder, J. A., and Katz, J. J., (Eds.) The Structure of Language, Prentice-Hall, Englewood Cliffs, N. J., 1964.

7. Green, B. F., Wolf, A. W., Chomsky, C., and Laughery, K. "BASEBALL: An Automatic Question Answerer" in Feigenbaum, E., and Feldman, J. (Eds.), Computers and Thought, McGraw-Hill, New York, 1963.

8. Minsky, M. L., "Steps Toward Artificial Intelligence," in Feigenbaum, E., and Feldman, J. (Eds.) Computers and Thought, McGraw-Hill, New York, 1963.

9. Quillian, M. R., "Semantic Memory" Ph.D. Thesis, Psychology Dept., Carnegie Institute of Technology, 1966 (available as BBN Report 1352).

10. Raphael, B., "A Computer Program which 'Understands'" AFIPS Conference Proceedings, Vol. 26, FJCC, 1964, pp. 577-589, Spartan Press, Baltimore, Maryland, 1964.

11. Simmons, R. F., "Answering English Questions by Computer: A Survey" Comm. A.C.M., Vol. 8, No. 1, Jan. 1965, pp. 53-70.

12. Thompson, F. B., DEACON Breadboard Summary, RM64TMP-9, TEMPO General Electric Co., Santa Barbara, Calif. March 1964.

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Bolt Beranek and Newman Inc. 50 Moulton Street Cambridge, Massachusetts 02138	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

PROBLEMS IN NATURAL COMMUNICATION WITH COMPUTERS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Scientific Interim Report No. 5

5. AUTHOR(S) (Last name, first name, initial)

Bobrow, Daniel G.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
August 1966	15	12

8a. CONTRACT OR GRANT NO. AF 19(628)-5065 ARPA No. 627 Amendment No.2 b. PROJECT AND TASK NO. 8668 c. DOD ELEMENT 6154501R d. DOD SUBELEMENT N/A	9a. ORIGINATOR'S REPORT NUMBER(S) BBN Report #1439 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFCRL-66-620

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
The work reported was supported by the Advanced Res. Projects Agency, dtd March 1965	Hq. AFCRL, OAR (CRB) USAF, L G Hanscom Field Bedford, Mass. 01730

13. ABSTRACT

This paper gives an overview of the problems involved in the construction of a computer-based question-answering system designed to interact with the user in English. The system is viewed as containing five distinct parts - a parser, a semantic interpreter, an information storer, an information retriever, and an English output generator. There is a need for extensive interaction among these subsystems, and between the subsystems and the user. Examples are given of the type of processing done by each subsystem, and the nature of the possible interactions. The syntactic analysis described is based on a Chomsky type of transformational grammar. The semantic store is characterized by a form of the predicate calculus, with additional algorithms for computation, and structures designed for fast access to relevant data.

DD FORM 1 JAN 64 1473

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
question-answering systems man-computer interaction natural language communication semantics information retrieval syntactic analysis						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through ________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through ________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through ________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.