

January 1971

THE APPLICATION OF THEOREM PROVING TO  
INFORMATION RETRIEVAL

by  
L. Stephen Coles

Artificial Intelligence Group  
Technical Note 51  
SRI Projects 7963 and 8696

The research reported herein was sponsored in part by the Lister Hill National Center for Biomedical Communications of the National Library of Medicine under Contract NLM-69-13, and in part by the National Science Foundation under Grant GN-895.

Presented at the 5th Hawaii International Conference on Systems Sciences, Honolulu, Hawaii, 1972.

## ABSTRACT

Most conventional computer information-retrieval systems are limited by rigid data structures and inflexible query languages. Computer question-answering systems designed to overcome either or both of these limitations have been built, but for the most part they have been restricted to small data bases. In this paper we will describe an approach to combining and extending recently developed question-answering techniques to reasonably large data files. A compilation of widely used physical laws and effects of interest to both engineers and scientists consisting of  $10^5$  basic data items will be used as a basis for demonstrating theorem-proving techniques on a large file. A restricted natural-language input for querying the file is also described.

## DESCRIPTIVE TERMS

Question answering, natural language, information retrieval, theorem proving.

## I INTRODUCTION

Thus far in the Artificial Intelligence Group at Stanford Research Institute we have attempted to apply theorem-proving techniques in the areas of question answering,<sup>1,2,3\*</sup> problem solving,<sup>4,5,6</sup> and reasoning by analogy.<sup>7</sup> In seeking to demonstrate these methods on interesting real-world problems, however, we have always encountered the same problem--the problem of a large data base. It soon became obvious that a finely tuned machine suitable for generating deep inferences based on a few relevant axioms was wholly inappropriate for coping with comparatively shallow inferences in the presence of large numbers of irrelevant axioms. The problem of logical deduction was converted from a problem of inference-making to a problem of "finding a needle in a haystack." Indeed, for "table-lookup" type queries that normally give rise to three-line proofs, we were discouraged to find that in terms of response time our supposedly powerful question-answering system was markedly inferior to conventional information-retrieval systems.

Figure 1 illustrates four inference problems as a function of size of data base and depth of inference required. Problems of Type 1 are essentially trivial and could be handled either by a theorem-proving or an information-retrieval system. Type 2 problems are naturally suited to a theorem prover, while Type 3 problems are well suited to an information-retrieval system. Type 4 problems, however, are not well suited to either approach. Yet they are most typical of the real world, being characterized by requirements for deep inferences based on data

---

\*References are listed at the end of this paper.

		Depth of Inference Required	
		Shallow	Deep
Size of Data Base	Small	Type 1 Input: John is a boy Query: Is John a boy?	Type 2 Input: Axioms of group theory Query: Is the order of a finite group divisible by the order of each one of its subgroups?
	Large	Type 3 Input: Airline reservation data Query: Is there a seat on today's flight from San Francisco to New York?	Type 4 Input: Back issues of the New York Times for ten years Query: What is U.S. foreign policy?

FIGURE 1 FOUR INFERENCE PROBLEMS

embedded in a large base of irrelevant facts. Therefore, we asked: Could the theorem-proving approach be combined with a conventional information-retrieval approach to consider problems in this fourth category?

Rather than trying to add a theorem-proving component to a conventional information-retrieval system, our strategy was to try to integrate information-retrieval capabilities into our theorem prover. Our first experience with this approach was the ENGDRG System,<sup>8,9</sup> developed in 1968-69 for medical drug-information retrieval. Our current work, the ENGLAW System, is concerned with a physical laws and effects data base<sup>10</sup> of interest to scientists and inventors. We will briefly describe these systems in turn. Finally, we will discuss some of the problems of implementing ENGLAW.

## II THE ENGDRG SYSTEM

ENGDRG evolved through two separate stages. The first phase involved a data base of drug descriptions for 25 anti-hypertensive drugs listed in the 1968 Edition of The Physician's Desk Reference (PDR). It contained approximately 300 binary relations in all, characterizing drugs in terms of chemical name, action, indications, contraindications, warnings, side effects, precautions, dosage, method of supply, and manufacturer. A typical interaction with this early version was as follows:\*

---

\* In this man/machine dialog, as well as those to follow, the human's typing is shown in both upper and lower case, and computer output is represented in all upper-case letters.

What is the action of halabar?

THE ACTION OF HALABAR IS SEDATION.

Is there an enzyme inhibitor?

YES, EUTONYL.

John Smith is a patient with moderate hypertension.

OK.

His symptoms include coronary artery disease.

OK.

Can you recommend a drug for John that does not cause headaches?

YES, SINGOSERP.

In formulating a correct reply for the question about John Smith, a hypothetical patient with moderate hypertension, ENGDRG had to inspect its data base to recall that of four possible drugs suitable for controlling moderate hypertension (aldomet, apresoline, singoserp, and unitensen), two of them (aldomet and apresoline) might possibly include headaches among their side effects and that unitensen, of the two remaining, was contraindicated because of the patient's coronary artery disease.

The subset of English used for querying the data base was highly restrictive, but suitable for this sort of dialog. The methods used to implement the natural-language interface are described in Refs. 11 and 12.

The second phase of our work on ENGDRG expanded to include the full range of drugs, but focused on the problem of drug/drug interactions. The data, approximately 150 three-place relations, were based on a table of

selected drug interactions by Morrelli and Melmon (Appendix A of Ref. 9). After six weeks of testing, this data base proved insufficient for the class of actual patient medication problems we wished to consider. Drs. Miller and Hirshman of the University of California Medical Center in San Francisco created an extended table more tailored to our needs (Appendix B of Ref. 9). Experimentation with the new data base, consisting of approximately 350 six-place and 50 binary relations, was carried on for about two months. One of those months was actually devoted to testing the data by remote teletype link to several doctors at the Drug Information Center in San Francisco.

A typical example of the use of ENGDRG with the Miller-Hirschman table might be as follows:

Does penicillin and aspirin produce an increased penicillin effect?

YES.

What is the clinical significance of the interaction between coumarin and barbiturate?

HOW MANY DAYS HAS THE PATIENT BEEN TAKING THE BARBITURATE?

8 days.

YES, IN THAT CASE THE INTERACTION IS SIGNIFICANT.

Can tetracycline interact with penicillin?

YES, TETRACYCLINE INTERACTS WITH PENICILLIN TO PRODUCE DECREASED PENICILLIN EFFECTS.

Does alcohol and barbiturate produce sedation?

I DON'T KNOW.

What are the interactions for tetracycline?

TETRACYCLINE INTERACTS WITH

- (A) PENICILLIN TO PRODUCE DECREASED PENICILLIN
- (B) TRIVALENT CATIONS TO PRODUCE DECREASED TETRACYCLINE
- (C) DIVALENT CATIONS TO PRODUCE DECREASED TETRACYCLINE.

What is the clinical significance of the interaction between salicylate and penicillin?

IT IS NOT SIGNIFICANT AT USUAL DOSES.

Warfarin is a coumarin.

OK.

Orinase is a sulfonylurea.

OK.

Can warfarin interact with orinase?

YES, TO PRODUCE DECREASED ORINASE.

What are the expected symptoms from the interaction?

THE EXPECTED SYMPTOM IS HYPOGLYCEMIA.

The latter part of the dialog illustrates how the pharmacist might add information about particular members of the general drug classes described in the data base and then confine his queries to these drugs. The average response time to queries on the SDS-940 computer implementation ranged from 30 seconds for simple questions to  $4\frac{1}{2}$  minutes for the most complex questions involving multiple references to the data base. Whenever replies to queries involved a mandatory reference to data contained on secondary storage (disk file), the response time was increased by an average of three minutes, thus illustrating the penalty of operating in this mode on the 940.

Encouraged by the fact that we were installing a new PDP-10 computer system approximately an order of magnitude faster than the 940 and also by

our preliminary success with the ENGDRG System, we planned a much more ambitious information-retrieval effort with a data base of  $10^5$  basic items concerning physical laws and effects.

### III THE ENGLAW SYSTEM

The physical laws and effects data base, of use to engineers, inventors, or laboratory scientists interested in a physical law or relation that will accomplish a desired effect, was taken from Hix and Alley.<sup>10</sup> As a first step in encoding the data base into axioms of the predicate calculus, we translated the 28-page Cross-Reference by Physical Quantity Index into approximately 700 binary relations. To give some idea of the current performance of ENGLAW on the PDP-10 using only the index part of our data base, consider the following hypothetical dialog:

Which laws concern friction?

THERE ARE THREE LAWS. DO YOU WANT TO SEE THEM?

Yes.

- (1) FRICTION EFFECTS
- (2) JOHNSON-RAHBK AND/OR WINSLOW EFFECT
- (3) TRIBOELECTRICITY.

What are the physical parameters of the Rahbek Effect?

CHEMICAL COMPOSITION, ELECTRIC FLUX, FRICTION, LIQUID,  
AND POTENTIAL.

The following hypothetical scenario illustrates how we expect a scientist or engineer might use ENGLAW on the complete file of 128 physical laws and effects, which are now in the process of being encoded:

1. I'm interested in a method for cooling. What are the relevant principles?

THERE ARE 9 PRINCIPLES THAT MAY BE RELEVANT. DO YOU WANT TO SEE THEM?

2. No. Among them are there any that involve electricity?

YES, THE GALVANOMAGNETIC-THERMOMAGNETIC EFFECTS AND THE THERMOELECTRIC EFFECTS.

3. Which one does not require the use of a magnetic field?

THE THERMOELECTRIC EFFECTS: THE SEEBECK EFFECT, THE PELTIER EFFECT, THE THOMPSON EFFECT, AND THE BENEDICK EFFECT.

4. Could one of these be used for building a refrigerator?

YES, THE PELTIER EFFECT.

5. Describe this effect.

WHEN A CURRENT FLOWS ACROSS THE JUNCTION OF TWO UNLIKE METALS, IT GIVES RISE TO A LIBERATION OR ABSORPTION OF HEAT. IF THE CURRENT FLOWS IN THE SAME DIRECTION AS THE CURRENT AT THE HOT JUNCTION OF A THERMOELECTRIC CIRCUIT OF THE TWO METALS, HEAT IS ABSORBED; IF IT FLOWS IN THE SAME DIRECTION AS THE CURRENT AT THE COLD JUNCTION OF THE THERMOELECTRIC CIRCUIT, HEAT IS LIBERATED. THE HEAT DEVELOPED IN A JUNCTION OF TWO METALS IS PROPORTIONAL TO THE FIRST POWER OF THE CURRENT AND DEPENDS ON THE DIRECTION OF THE CURRENT.

THE QUANTITY OF HEAT ABSORBED OR LIBERATED IS GIVEN BY THE FOLLOWING EQUATIONS:

$$\text{RATE OF HEAT ABSORBED} = IT(DE/DT),$$

WHERE I = CURRENT (A)

T = ABSOLUTE TEMPERATURE ( $^{\circ}$ K)

DE/DT = RATE OF CHANGE OF THERMO EMF WITH TEMPERATURE.

THE RATE OF HEAT LIBERATED AT THE JUNCTION WILL BE

$$W = I^2 R_J + IT_0(SA-SB)$$

WHERE W = WATTS AT THE JUNCTION

I = JUNCTION CURRENT FLOW FROM A TO B

T<sub>0</sub> = TEMPERATURE AT THE COLD END

R<sub>J</sub> = JUNCTION RESISTANCE

SA,SB = ENTROPY TRANSPORT FACTORS OF MATERIALS A AND B.

6. What will the rate of heat liberated at the junction be if  $I = 2A$ ,  $R_J = 1000$  ohms,  $T_0 = 270^\circ K$ ,  $SA = 10$ , and  $SB = 5$ ?  
6,700 WATTS.
7. What is the efficiency of this effect?  
LESS THAN ONE PERCENT.
8. Where can I learn more about the Peltier Effect?  
W. C. WHITE, "SOME EXPERIMENTS WITH THE PELTIER EFFECT,"  
ELECTRICAL ENGINEERING, JULY 1951.

The above scenario illustrates many important features of ENGLAW, including its information retrieval, deductive, computational, and English input/output capabilities. The basic approach, as with ENGDRG, is to

- (1) Translate English input statements, questions, and commands into a formal language based on the first-order predicate calculus.
- (2) Determine which axioms are relevant, if any.
- (3) Perform any necessary deductive inferences or computations based on the axioms.
- (4) Generate an appropriate reply in English or display a piece of text from the data base, leading ultimately to another input from the human.

#### IV IMPLEMENTATION OF ENGLAW

The overall system can be viewed as consisting of seven components:

- (1) A syntax analyzer for simple English declarative, interrogative, and imperative sentences.
- (2) A semantic interpreter that maps the output of the syntactic component into a deep structure representation based on the first-order predicate calculus.

- (3) A deductive component using the QA3.5 theorem-proving system.
- (4) An information-retrieval component for immediate table look-up operations.
- (5) An axiom model containing both facts and relations about the universe of discourse in predicate calculus form,
- (6) A text data base of the source materials held as English sentences stored on disk file.
- (7) A sentence generator for outputting replies in English.

The initial translation to the predicate calculus is accomplished by means of productions in the syntactic and semantic components. Syntactic productions are essentially rewrite rules from a "bottom-up" syntax analyzer based on the phrase structure grammar specifying the fragment of natural language chosen for communication. Semantic productions, on the other hand, are rewrite rules operating on well-formed formulas (wff) in the predicate calculus corresponding to the meaning of the current contents of the syntactic stack. If the syntactic analysis terminates successfully, then the output of the last semantic production executed corresponds to the meaning of the original sentence. More details on the operation of the syntactic and semantic components can be found in Refs. 11 and 12.

Logical inferences in the deductive component are carried out by means of the QA3.5 resolution-based theorem prover. The resolution principle is a rule of inference especially suited to theorem proving by computer. The negation of the wff is first put into a standard form called Prenex Normal Form. In this form a wff is represented as the

conjunction of a set of formulas called clauses, each of which is a disjunction of elementary formulas called literals. Using resolution, new clauses are deduced from the starting clauses. The goal of the procedure is to deduce a contradictory formula of the form  $\phi \wedge \sim\phi$ , demonstrating that the starting formula is contradictory, and thus the unnegated original formula is a theorem. More information on the method of operation of the deductive component can be found in Refs. 1, 2, and 4.

The information-retrieval component consists of a set of rapid-access template-matching routines especially effective for finding all references to some axiom pattern in the data base rather than just the first one that satisfies it. These have been tested already on the Cross-Reference Index data base.

Determining which axioms are relevant among the thousands of axioms in the data base is a crucial step, since inference should not be attempted until the potentially useful axioms have been narrowed as much as possible. The search for relevant axioms will be accomplished at several levels, taking advantage of the hierarchical structure of the data.

English output sentences are produced by translating answer expressions in the predicate calculus into their English equivalents, again by means of a set of productions. This method is briefly described in Ref. 12. The original English text of the data base will be kept out on disk as the most convenient way of replying to queries of the form "Describe such and such a law or effect." It is reasonably well formatted,

with each of the 128 basic law or effect categories being divided into five sections:

- (1) A basic description
- (2) An illustration of its principal use
- (3) A brief discussion of its magnitude limitations
- (4) Reference to the literature
- (5) A set of descriptive terms.

To demonstrate in a general way the feasibility of conducting the dialog presented in the previous section, let us consider how Questions 4 and 7 might be answered. First we must translate these questions from English into the predicate calculus. Q4 might be translated by the syntactic and semantic components as

Q4:  $(\exists x,y) \{ (Is(x,SE) \vee Is(x,PE) \vee Is(x,BE)) \wedge Is(y,refrigerator) \wedge Usedtobuild(x,y) \}$

and Q9 might be translated as

Q7:  $(\exists x,y) \{ Is(x,PE) \wedge Efficiency(x,y) \}$  .

The section of English text taken from the data base (page 217 of Ref. 10) relevant to these questions is as follows:

"In actual tests, the Peltier effect efficiency has been found to be less than 1%. The best that can be done by use of ordinary metals is to cool a small bit of metal by not more than 10°C. These figures have been improved upon by the use of semi-conducting materials so that temperature differences of 20°C have been obtained. These semiconducting materials have been used to build experimental refrigerators."

Of all the problems in implementing ENGLAW, one of the most difficult is the building up of an axiom model in predicate-calculus form based on

the original English text. The essence of the problem is the proper choice of predicates, connectives, and quantifiers for representing English facts in an axiomatic form. Clearly many possible translations to predicate calculus would serve equally well in capturing the meaning of the sentence. Yet, not all of these would lend themselves to efficient theorem proving.

The ideal solution, of course, would be to carry out the translations from English into predicate calculus automatically using the already existing syntactic/semantic components of ENGLAW. The problem with this approach is that the English query language used by the human for interrogating the data base is only a small fragment of natural language. The syntactic/semantic components of ENGLAW were never designed to handle the full range of English such as appears in the text of the data base. Indeed, the sort of grammatical constructions and anaphoric references that typically appear in text are far more complex than can be handled by any existing computer translation system. Thus, in order to use these components at all in solving this problem one would first have to translate the original English text manually into a simplified subset of English. Because this problem is itself nontrivial, we decided to translate the text directly to predicate calculus by hand. We are still in the process of developing guidelines for such manual translation.

The axioms derived from the section of the data base shown above are as follows:

A1:  $(\exists x, y, z) \{ \text{Is}(x, \text{PE}) \wedge \text{Efficiency}(x, y) \wedge \text{Less}(y, 1\%) \wedge \text{Found}(y, z) \wedge \text{Is}(z, \text{actual}) \wedge \text{Is}(z, \text{test}) \}$

- A2:  $(\forall x,y)\{Is(x,PE) \wedge Is(y,ordinary) \wedge Is(y,small) \wedge Is(y,bit) \wedge Is(y,metal) \Rightarrow (\exists z)\{Temperature(y,z) \wedge Less(d(z),10^{\circ}C)\}\}$
- A3:  $(\forall x,y)\{Is(x,PE) \wedge Is(y,semiconductor) \wedge Is(y,material) \Rightarrow (\exists z)\{Temperature(y,z) \wedge Equal(d(z),20^{\circ}C)\}\}$
- A4:  $(\exists x,y)\{Is(x,semiconducting) \wedge Is(x,material) \wedge Usedtobuild(x,y) \wedge Method(x,PE) \wedge Is(y,experimental) \wedge Is(y,refrigerator)\}$
- P1:  $(\forall x,y,z)\{Method(z,x) \wedge Usedtobuild(z,y) \Rightarrow Usedtobuild(x,y)\}$

The deductive component, QA3.5, would then make use of these axioms to develop the answer predicates:

- Q4: Yes,  $x = PE$ , based on P1 and A4, and
- Q9: Yes,  $y = \text{Less than } 1\%$ , based on A1.

The English generative grammar would then produce the correct English response for each reply.

The advantage of the first-order predicate calculus representation should now be clear. By the proper addition of principles to the fundamental axioms based on facts stated in the data base, the class of answerable questions becomes considerably broader.

## V CONCLUSION

In the course of implementing ENGLAW, a wide variety of problems and difficulties arose--hardware, software, linguistic, logical, response time, and so on. There are still many questions that remain to be answered before a deductive information-retrieval system will be both practical and effective in the logical analysis of data. Among them are the following:

- (1) What should be the structure of multilevel storage devices for providing both the capacity demanded by a large data base and the response time needed for conversational interaction?
- (2) What heuristics could best be applied to a resolution theorem prover attempting to make deductions in the presence of large sets of irrelevant premises?
- (3) What should be the scope of a habitable query language suitable for the nonspecialist user?

In approaching the first of these questions we are beginning to explore the use of a hash-coded index in core memory to data held on disk. We will also look into the problem of multiple copies of the data base differently indexed by predicates and arguments as suggested by Levien and Maron.<sup>13</sup> In exploring the second question we will try to follow the suggestion of Darlington<sup>14</sup> in trying to maximize the predicate diversity of the data base so as to fully capitalize on the set of support and ordering heuristics inherent in the theorem prover.

Our experience with the third question so far has forced us to reconsider the flexibility of the man/machine dialog. The system must estimate in advance the amount of information it is prepared to supply the user in response to his question. This gives the user the option to decide whether he really wants all of it. The system must also estimate how long it will take to answer a question in order to provide the user with additional options. The experienced user, on the other hand, has already cultivated a good intuition about the demands he

normally makes on the system and is indignant about the thought of paying the response-time overhead costs associated with the extra generality he doesn't need. This places an extra requirement for a modular query language interface on ENGLAW.

The problem of plausible inference based on uncertain data is one that we have not even considered in this paper. Yet, associated with most data from the real world is this extra feature of unreliability. To begin approaching the practical problems of users from the intelligence community, having large data bases of uncertain facts, wholly new theoretical tools like fuzzy sets<sup>15</sup> may be needed.

In spite of these many unanswered questions, it is felt that ENGLAW will illustrate the feasibility of integrating a theorem prover with information-retrieval techniques in the logical analysis of a large body of data. The success of this approach in practical applications, however, must still be demonstrated.

#### ACKNOWLEDGMENTS

The drug interaction data base (Appendix B of Ref. 9) was made possible through the painstaking efforts of Drs. Robert A. Miller and Joseph L. Hirschman of the Drug Information Center of the University of California Medical Center in San Francisco. They both acted as consultants to our project in the matter of drug/drug interactions and regarding the operation of the center. The author would also like to express his appreciation to Oliver Whitby, Bertram Raphael, and Charles Rosen for their many suggestions. Mrs. Ann Robinson is responsible for much of the software implementation.

## REFERENCES

1. C. C. Green and B. Raphael, "The Use of Theorem-Proving Techniques in Question-Answering Systems," Proc. 23rd National ACM Conference, pp. 169-181 (Brandon Systems Press, 1968).
2. C. C. Green, "Theorem Proving by Resolution as a Basis for Question-Answering Systems," in Machine Intelligence 4, Meltzer and Michie, Eds. (Edinburgh University Press, Edinburgh, Scotland, 1969).
3. L. S. Coles and C. C. Green, "Chemistry Question-Answering," Technical Note 9, Artificial Intelligence Group, Stanford Research Institute, Menlo Park, California (June 1969).
4. C. C. Green, "Application of Theorem Proving to Problem Solving," Proc. First International Joint Conference on Artificial Intelligence, Washington, D.C., pp. 219-239 (The Mitre Corporation, Bedford, Massachusetts, May 1969).
5. L. S. Coles, "An Experiment in Robot Tool Using," Proc. IEEE Systems Science and Cybernetics Conference, Pittsburgh, Pennsylvania, October 14-16, 1970, pp. 224-225.
6. N. J. Nilsson, Problem-Solving Methods in Artificial Intelligence, (McGraw-Hill Book Company, New York, New York, 1971).
7. R. E. Kling, "Reasoning by Analogy as an Aid to Heuristic Theorem Proving," Technical Note 49, Artificial Intelligence Group, Stanford Research Institute, Menlo Park, California (submitted to IFIP Congress, 1971).
8. J. H. Chadwick et al., "Medical Applications of Remote Electronic Browsing," Final Report, SRI Project 7382, Stanford Research Institute, Menlo Park, California (February 1969).

9. L. S. Coles et al., "Design of a Remote-Access Medical Information-Retrieval System," Final Report, Contract NLM-69-13, SRI Project 7963, Stanford Research Institute, Menlo Park, California (November 1969).
10. C. F. Hix and R. P. Alley, Physical Laws and Effects (John Wiley and Sons, Inc., New York, New York, 1958).
11. L. S. Coles, "An On-Line Question-Answering System with Natural Language and Pictorial Input," Proc. 23rd National ACM Conference, pp. 157-167 (Brandon Systems Press, 1968).
12. L. S. Coles, "Talking with a Robot in English," Proc. First International Joint Conference on Artificial Intelligence, pp. 587-596 (The Mitre Corporation, Bedford, Massachusetts, 1969).
13. R. E. Levien and M. E. Maron, "A Computer System for Inference Execution and Data Retrieval," Comm. ACM, Vol. 10, No. 11, pp. 715-721 (November 1967).
14. J. L. Darlington, "Theorem Proving and Information Retrieval," in Machine Intelligence 4, Meltzer and Michie, Eds., pp. 173-181 (Edinburgh University Press, Edinburgh, Scotland, 1969).
15. L. A. Zadeh, "Fuzzy Sets," Information and Control, Vol. 8, No. 3, pp. 338-353 (June 1965).