MEDCAT: An interactive computer program for medical diagnosis, consultation, and teaching

by W. D. HAGAMEN, MARTIN GARDY, GREGORY BELL, EDWIN REKOSH, and STEVEN ZATZ
Cornell University Medical College
New York, New York

ABSTRACT

MEDCAT is a computer program that makes diagnoses, explains each step in its reasoning in response to questions, increases its knowledge and reasoning ability by conversing with expert physicians, and uses its logical and communicative skills to help medical students in the proper approach to medical diagnosis and to evaluate their progress in this area. The mechanism for each of these features is discussed. MEDCAT is coded in APL and implemented on a 68000-based microcomputer.
INTRODUCTION

Patient populations on medical floors of teaching hospitals are heavily skewed toward the seriously ill and the previously diagnosed. As a result, the variety of diagnostic problems encountered is limited. The reasons for this are not relevant here, but one of the generally recognized consequences is that medical students do not have enough opportunity to develop their clinical reasoning. The primary purpose of MEDCAT is to help fill this void by providing students with the opportunity to sharpen their diagnostic skills through interaction with a computer program that contains a carefully selected range of patient records.

It was necessary to give MEDCAT four specific capabilities to achieve its goal. First, MEDCAT arrives at its own diagnoses by using the empirical data contained in a patient's medical record and its knowledge of medical logic. Second, it can explain any step in its reasoning in response to questions. Next, it chooses appropriate strategies and takes the initiative in asking questions as it guides and evaluates the student's performance. Finally, it adds to its own knowledge and reasoning ability by free-format discussions with expert physicians.

The essence of MEDCAT's intelligent behavior is found in two areas: the method used to represent both its medical knowledge and logic as data external to the code, and the way these data are interfaced with the external world through its natural language abilities.

Because we wish to illustrate not only what the program's function is, but also how it works, we must use some medical terms and concepts that may be unfamiliar. However, we simplify the diagnostic logic to illuminate the explanation of the process itself. MEDCAT is coded in APL and implemented on a 68000-based microcomputer.

REPRESENTING MEDICAL KNOWLEDGE AND LOGIC

The data in the program are viewed as a network of nodes and pointers. Figure 1 shows 8 nodes (boxes and circles) and the 11 pointers (arrows) that connect them. These pointers may be either positive (facilitory) or negative (inhibitory). Facilitory pointers are shown as solid arrows; inhibitory pointers are dashed arrows. Each pointer is associated with a numeric value.

Node Descriptors

Each node has a descriptor (noun phrase), which is used by the program to identify the subject of a question and to generate its response. These node descriptors are shown in abbreviated form in Figure 1. The complete node descriptors, together with their node numbers, are as follows:

156 IGM antibody to hepatitis-A virus (IGM AB-HAV)
157 IGM antibody to hepatitis-A virus (IGM AB-HAV)
158 Hepatitis-A (acute)
165 Hepatitis-B surface antigen (HBSAG)
166 Hepatitis-B surface antigen (HBSAG)
167 Hepatitis-B (acute)
168 Hepatitis non-A non-B (acute)
345 Viral hepatitis (acute)

Figure 1—Schematic representation of medical logic
Each node also has five additional attributes, each of which is stored as a vector (Table I). The node profile is an integer indicating what the node represents. The first six represent empirical data (1, chief complaint; 2, demographic data; 3, history; 4, symptoms; 5, signs; and 6, laboratory tests). Profile types 7–11 represent diagnoses with varying levels of specificity—11 is the most specific. Nodes 156, 157, 165, and 166 are laboratory tests. The remaining are diagnostic nodes. The three with profiles of 11 are more specific than viral hepatitis (10)—they are specific forms of viral hepatitis.

Table I—Numeric attributes

<table>
<thead>
<tr>
<th>Profile</th>
<th>Adjective</th>
<th>Threshold</th>
<th>Patient record</th>
<th>Stimulus</th>
</tr>
</thead>
<tbody>
<tr>
<td>156</td>
<td>6</td>
<td>100</td>
<td>NEG</td>
<td>1</td>
</tr>
<tr>
<td>157</td>
<td>6</td>
<td>100</td>
<td>POS</td>
<td>0</td>
</tr>
<tr>
<td>158</td>
<td>11</td>
<td>500</td>
<td>NEG</td>
<td>1</td>
</tr>
<tr>
<td>165</td>
<td>11</td>
<td>500</td>
<td>POS</td>
<td>0</td>
</tr>
<tr>
<td>166</td>
<td>10</td>
<td>500</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>167</td>
<td>10</td>
<td>500</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>168</td>
<td>10</td>
<td>500</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>345</td>
<td>10</td>
<td>500</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

Numeric Attributes

The necessary information about each pointer is stored in a column of a three-row matrix. Row 1 indicates the node from which the pointer originates and row 3 indicates the node to which it points. Row 2 contains the value of the pointer. That part of the pointer matrix corresponding to Figure 1 is shown in Table II.

CONSULTATION MODE

The physician or student selects a patient. The program retrieves the patient record and executes the diagnostic subroutine. This takes about 3 seconds. The physician may then ask the computer questions about the reasoning it used to evaluate the patient's problem:

**Physician:** Did you also think of hepatitis-A?
**Computer:** Yes, hepatitis-A was considered, but rejected.
**Physician:** What made you reject it?
**Computer:** Because the negative IGM antibody to hepatitis-A virus argued strongly against hepatitis-A.

We will put aside for the moment the question of how the program knew which node was being asked about and focus only on how it applied the available data (logic) to answer the question. The "why" nature of the question caused it to look at the facilitory pointers to hepatitis non-A non-B and to the diagnosis of viral hepatitis confirmed hepatitis non-A non-B. Table II—Pointer matrix

<table>
<thead>
<tr>
<th>From</th>
<th>156</th>
<th>156</th>
<th>157</th>
<th>157</th>
<th>165</th>
<th>165</th>
<th>165</th>
<th>166</th>
<th>166</th>
<th>166</th>
<th>345</th>
<th>345</th>
<th>345</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>To</td>
<td>158</td>
<td>158</td>
<td>158</td>
<td>158</td>
<td>167</td>
<td>167</td>
<td>167</td>
<td>167</td>
<td>168</td>
<td>168</td>
<td>168</td>
<td>168</td>
<td>168</td>
</tr>
</tbody>
</table>
The program says that hepatitis-A was considered because there is at least one active facilitatory afferent pointer going to it. It says hepatitis-A was rejected because the node was not turned on. The reason for rejection was obtained by looking at the active inhibitory pointer.

**Physician:** Why the IGM AB-HAV?
**Computer:** To determine the likelihood of hepatitis-A and hepatitis non-A non-B.

When the “why” question refers to an empiric node, the program looks to the efferent pointers from the node to find out what decision (diagnostic node) these pointers would influence.

**STUDENT MODE**

When medical students use MEDCAT to practice their diagnostic skills, they experience three phases of the program in sequence.

**Phase I**

Aside from a brief narrative summary generated by the program, the student obtains patient information only by asking questions. The program keeps track of these questions and will use this information later to guide its questioning of the student’s reasoning.

When a student tries to obtain demographic, historic, or symptomatic information, he assumes he is speaking directly to the patient. This permits the program to evaluate and comment on interviewing skills. A common error of beginning students is to ask questions that influence a patient’s answer.

**Student:** Did you have pain below the ribs on the right side?

At this point the program will break in with the following comment and require the student to ask another question:

**Computer:** That is a leading question.
**Student:** Have you had any pain?
**Patient:** Yes.
**Student:** Where was the pain?

If the question the student asks the patient has a syntax similar to “did you have” or “was there,” and if it matches only one node in the list of node descriptors with profile types 2-4, it is treated as a leading question. If it matches on more than one, it is accepted. There is only one node descriptor that contains the words “below pain ribs right side.” There are many that contain only “pain.” The patient gives an affirmative answer if it is true for any of the matches—otherwise the patient answers negatively. Questions with syntax such as “where was the pain?” or “what was its quality?” are not considered leading questions.

The student also has the option of asking ELIZA-like questions. This is considered good interviewing technique.

**Phase II**

When the student finishes asking for patient data, the program takes the initiative. By this time it has made its own evaluation of the patient’s problem. It also knows what questions the student asked and, therefore, what conclusions the student may or may not draw. The program chooses a strategy based on these two sets of data. The program begins by asking about the diagnosis:

**Computer:** What was your diagnosis?
**Student:** Hepatitis non-A non-B.

If the student’s answer agrees with the program’s own evaluation, as happens here, the program determines whether the reasoning was sound:

**Computer:** Why did you diagnose hepatitis non-A non-B?
**Student:** Because the IGM AB-HAV and HBSAG were negative.
**Computer:** How do you know he had liver disease?
**Student:** Because there was evidence of acute hepatitis.
**Computer:** How did you rule out toxic hepatitis?
**Student:** There was no history of exposure to toxins or drugs.

IGM AB-HAV and HBSAG are recognized abbreviations for IGM antibody to hepatitis-A virus and hepatitis-B surface antigen, but so far the student’s reasoning is incomplete. We would want to know whether viral hepatitis was part of his logic. Viral hepatitis and acute hepatitis are not synonymous. Acute viral hepatitis is one form of acute hepatitis and is, therefore, more specific (profile type 10 vs. type 9). However, we do not want to teach students to parrot answers. This student is doing fine. The program assures itself that the stu-
dent is on the right track, without mentioning viral hepatitis, by asking why he rejected another type of acute hepatitis. The program will continue asking questions such as what other diagnoses were considered, why they were considered, and why they were rejected. It keeps a running score of the student's performance, and when the student is doing this well, it terminates the discussion after three or four major topics have been explored.

Other strategies are dictated by the student's answers to the initial and subsequent questions. First, if the student gets the right diagnosis, but for the wrong reasons, the program pursues this vigorously. Next, if the student were to give hepatitis as the diagnosis, the program would ask questions designed to lead up to the more specific (etiologic) diagnosis. Finally, if the student does not have sufficient knowledge of the patient's data because the questions were not asked in Phase I, these findings are presented to the student at the appropriate time.

**Phase III**

During Phase I the students are allowed to ask for empirical data only. They may not ask questions about diagnoses. The program asks the questions during Phase II. Phase III permits the students to resolve any questions that may have arisen or that were unanswered in previous phases. It is identical in format to the consultation mode.

**LEARNING MODE**

New knowledge and logic can be added to the program by free-format discussions with qualified experts. In the following example the program has made an erroneous diagnosis. It must correct itself, learn about a new disease, and find the basis for a new diagnosis. This further illustrates the usefulness of the system's method of data representation and its language capabilities.

**Physician:** What was your diagnosis?
**Computer:** Hepatitis non-A non-B.
**Physician:** That's wrong.
**Computer:** Why is it wrong?
**Physician:** Because the positive beef red cell absorption test excluded it.
**Computer:** Why did you order the beef red cell absorption test?
**Physician:** Because the heterophil agglutinin test was positive.
**Computer:** Why did you order the heterophil agglutinin test?
**Physician:** Because there were enlarged cervical lymph nodes and a flu-like syndrome.

The data nodes referred to in this discussion already exist. The program is simply adding pointers to the pointer matrix (Table II). The physician's answer to why it was wrong caused a new (inhibitory) pointer to be added from an existing node (38), called "beef red cell absorption," to "hepatitis non-A non-B" (168). The program chose node 38 over node 37, which has the same descriptor, because 38 has a stimulus value of 1. Verification is done by comparing its adjective value with the adjective used by the physician. The verb "excludes" causes the pointer to be sufficiently negative to outweigh all facilitory pointers. This is accomplished by adding a column (38, -1100, 168) to the pointer matrix.

The reason given for ordering the beef absorption test added a pointer with a value of 1. Such weak pointers to empiric nodes affect only the conversational phases of the program. These are discussed when we describe how the threshold is calculated.

The program now understands why its diagnosis was incorrect, but it does not know what the diagnosis should be. It asks:

**Computer:** What is the diagnosis?
**Physician:** Infectious mononucleosis.
**Computer:** Why did you diagnose infectious mononucleosis?
**Physician:** Because the beef absorption and heterophil agglutinin tests were positive.
**Computer:** Do you mean beef red cell absorption test?
**Physician:** Yes.
**Computer:** Were there any other reasons?
**Physician:** Enlarged cervical lymph nodes and flu-like syndrome.
**Computer:** Any others?
**Physician:** No.
**Computer:** How important (on a scale of 1-5) is the beef red cell absorption test?
**Physician:** 5.

Infectious mononucleosis represents a new node that is added to the list of node descriptors. It is a diagnostic node and, because it is replacing hepatitis non-A non-B, it is assumed to have the same profile type. Because it is a diagnostic node, place-holders are required only for its adjective type, stimulus value, and patient record. Its threshold, as with any node, is calculated prior to execution.

Four pointers to infectious mononucleosis from the four nodes indicated must be added. Because the physician did not use any quantitative verbs or adverb-verb combinations to indicate pointer values, these are explicitly requested. This process is repeated for the remaining pointers. To test the results of this learning session, the diagnostic subroutine must be executed again:

**Physician:** What was your diagnosis?
**Computer:** Infectious mononucleosis.
**Physician:** Why?
**Computer:** Because the enlarged cervical lymph nodes, the positive beef red cell absorption and heterophil agglutinin tests, and the diagnosis of flu-like syndrome confirmed infectious mononucleosis.
**Physician:** Why did you order the beef absorption?
**Computer:** Because of the possibility of infectious mononucleosis, and because the heterophil agglutinin test was positive.
**Physician:** Why did you reject non-A non-B?
**Computer:** Because the positive beef red cell absorption test excludes it.
**Evaluating the Logic**

The program checks for obvious inconsistencies. It would reject "because the heterophil test was negative" because this disagrees with the patient record. Aside from conflicts with empirical data, however, the program will accept any line of reasoning as long as it can be made consistent with what it already knows. The program stops asking questions and accepts the logic when afferent pointers exist to each new diagnostic node.

**Calculating the Threshold**

The thresholds of new nodes need not be set by the physician. The threshold of all empiric nodes is set arbitrarily at 100. The values of afferent pointers to empiric nodes can be either $-1$ or $+1$. The algebraic sum of such pointers never exceeds the threshold of the empiric node and therefore cannot turn it on. Whether or not an empiric node is activated is determined solely by the patient data. This is an important theoretical point. Certainly one would not want the answer to a question or a test to be affirmative simply because the question was asked or the test was done.

These weak pointers are used when the program is generating answers to questions and when it evaluates a student's reasoning. The beef red cell absorption test was ordered because the heterophil agglutinin test was positive. As indicated earlier, we are particularly concerned that the student have adequate reason for requesting laboratory tests. When requesting information from experts, the program asks for reasons for laboratory tests, although it does not ask for reasons for other types of empiric nodes.

The program calculates the thresholds for diagnostic nodes. The basic process consists of dividing the sum of positive afferent pointers to the node by two and using this quotient or 500, whichever is greater, as the threshold. The sum is divided by two because we want to mimic varying degrees of certainty in a diagnosis. When the algebraic sum of afferent pointers to a node equals or barely exceeds the threshold, this is interpreted as a tentative diagnosis or working hypothesis. If it is more than twice the threshold, the diagnosis is confirmed.

Only pointers from active empiric nodes are used. This is necessary because each node in the set that represents a test has a different pointer value. One would not want to add these together when only one of them can be relevant. The use of pointers from active empiric nodes also is desirable because it permits the flexibility one sees in observing physicians at work. If a specific test or piece of information is not available, a given diagnosis should not be ruled out. A positive beef red cell absorption test may confirm infectious mononucleosis, but the presumptive diagnosis can be made without it.

Afferent pointers from diagnostic nodes are counted, whether active or not, because there is no way of determining this before executing the diagnostic subroutine. This is consistent with the role intermediate diagnoses should play. If an intermediate diagnosis is missing, the more specific diagnosis often should not be made.

**NATURAL LANGUAGE PROCESSING**

On the input side the basic tasks the program faces are to determine the purpose of the question and the nodes that are being referred to and (particularly in the learning mode) to isolate certain adjectives and verbs with quantitative implications. This must be done within a reasonable response time (2–3 seconds). Each task is simplified because of the limited verbal domain in which we work.

**Word Types and Parsing**

The sentence is first parsed into functional units or phrases. This is done by checking each word against a list of 128 keywords; each keyword is associated with a numeric value. Thus, a numeric vector is established with a value (word type) for each word in the sentence. These numbers indicate several things, including whether the associated word (or punctuation) should start or end a phrase; whether it is a recognized adjective, verb, or profile word; and whether it is negative or positive. 

"Excludes" encodes to $-6.9$. The 6 indicates that it is a verb that begins a phrase unless preceded by another verb or adverb. The minus sign indicates that it requires a negative pointer. The .9, together with its sign, indicates that the negative pointer must be strong enough to prevent firing of the node.

The program breaks up the sentence fragment, "because the positive beef red cell absorption test excludes it," into four separate phrases:

1. Because
2. The positive beef red cell absorption test
3. Excludes
4. It

The words in the keyword list are italicized for illustrative purposes. "The" is one of the words that flags the start of a noun phrase. A verb such as "excludes," unless preceded by an adverb or helping verb, signals the beginning (and end) of a verb phrase. The remaining italicized words are not needed for parsing in this instance.

It is important that the keyword list contain no nouns except profile words. The program uses the first word to characterize a phrase. If this word happens to be an article, as above, the phrase is clearly a noun phrase. However, articles often are omitted by students, particularly when they ask for lists of laboratory results. Commas, verbs, pronouns, and conjunctions signal the end of one phrase and the start of another. Because the keyword list contains most of the non-noun words that are used in our limited domain, the program may assume that a phrase is a noun phrase if it does not begin with a keyword.

**Node Selection**

The list of descriptors is treated as a string and is searched for the occurrence of single words in the input sentence. One purpose of parsing is to eliminate unnecessary words from the
search. Not only does this narrow the process to the noun phrase, but the italicized words in the example phrase are deleted before searching begins. This preprocessing maintains the necessary response time even though the descriptor list contains 30,000 characters and is growing.

In the consultation and student modes, the program picks the node with the highest number of matching words. In a previous example, "beef absorption" matched "beef red cell absorption." This is sufficient for a presumptive match, but because new nodes and pointers are being added, the program asks for verification in the learning mode: "Did you mean beef red cell absorption?"

In the string search, input words are preceded but not followed by a space. For this reason, "nucleosis" will not match with "mononucleosis," but "mono" will. This permits users to follow the inclination to abbreviate words.

When the question contains neither a subject nor a pronoun, "why?" is treated as "what" questions. Different tenses of the same verb are equated.

Enclosing these data (nodes, pointers, adjectives, profile) in appropriate narrative prose is simply a matter of applying flexible templates. The magnitude by which the algebraic sum exceeds the threshold determines the choice of verb phrases ("confirms," "strongly suggests," "suggests," or "is consistent with").

Dlscussion
MEDCAT has certain features in common with MYCIN. It has an instructional mode, it exhibits knowledge acquisition, and it can display reasons for its decisions. However, the single feature that most clearly distinguishes MEDCAT from MYCIN and other rule-based artificial intelligence systems is the representation of both its knowledge and its logic as a numeric matrix that serves a variety of functions. MEDCAT uses the matrix to arrive at its diagnoses, and the simple parser and text generator permit the program to interface with the user. This representation also accounts for the ease with which knowledge and logic can be evaluated and changed during free-format discussions. APL, which is often described as an array-processing language, is well suited to this purpose.

The learning that MEDCAT exhibits addresses the general problem of getting the most qualified experts to contribute to the logic and knowledge base. Experts usually are unaccustomed to analyzing the essential mechanisms that may underly their particular intellectual skills. They are experienced, however, in making decisions and justifying them once they are made. This is exactly what MEDCAT does—it engages experts on familiar intellectual turf. By asking questions such as "Why do you disagree?" or "Why do you think that?" or "Why did you order that test?" the program requires no more of physicians than what is expected of them on their teaching rounds.
ACKNOWLEDGMENTS

This work was supported in part by grants from the Frances L. and Edwin L. Cummings Memorial Fund and the Anonymous Gift Fund of Cornell University Medical College.

REFERENCES
