GENERAL

Domain

Medicine – Bacterial Infection

Main General Function

Diagnosis and Advice

System Name

MYCIN (and later, EMYCIN)

Dates

MYCIN: First publication 1974
EMYCIN: First publication 1980

Rough period of development

MYCIN: 1972-1978, EMYCIN 1974-1980

Researchers

MYCIN: Bruce Buchanan, Stanley Cohen, Edward H. Shortliffe (PhD thesis)
EMYCIN: William van Melle (PhD thesis), Shortliffe, Buchanan

Main people responsible for R&D

The work was performed within the Stanford Heuristic Programming Project, in collaboration with the Infectious Diseases Group at Stanford Medical School. The core group is described as Shortliffe, Buchanan, Davis, Scott, Clancey, Fagan, Aikens, and van Melle. Carli Scott was the first employee. Shortliffe, Cohen, Thomas Merigan and Stanton Axline provided the domain expertise.

Location

Stanford University, CA

Language

LISP (Interlisp)

Machine

DEC PDP-10

Brief Summary

MYCIN provides consultations to doctors on bacterial infections of the blood. It attempts to diagnose which bacteria may be involved, and it suggests appropriate drug treatments. It is designed with real use in mind, and so it requires little training on the user’s part, it is fast, it deals with uncertainty and incompleteness, and it can answer questions about its reasoning.

MYCIN was developed in part as an AI research effort, but mainly to help solve the problem of misdiagnosis and inappropriate drug therapy.

The name is not an acronym; “mycin” is a common prefix for antibacterial agents.

EMYCIN is an expert system shell or kernel derived from MYCIN. It introduced a number of improvements and cleanups. MYCIN was then re-implemented in EMYCIN.

EMYCIN stands for “Essential MYCIN.”

Related Systems

Strictly speaking, MYCIN was an original effort and was not developed from any other system. However, DENDRAL, which many consider the first expert system, was developed at Stanford before MYCIN and MYCIN’s authors acknowledge a debt to DENDRAL’s use of production rules with domain specific knowledge. Also, the MYCIN team collaborated on the MEDIPHOR drug-interaction warning system immediately prior to MYCIN.

EMYCIN was developed from MYCIN by some of the same group. The purpose of EMYCIN was to explicitly separate domain knowledge from MYCIN, to make it easier to build expert systems in other domains. EMYCIN included TEIRESIAS.

A number of systems have been built along the same lines. In a broad sense, MYCIN is the embodiment of “all the clichés of what expert systems are.” (Newell) In a stricter sense, its descendents are: EMYCIN as a base system with PUFF, SACON, CLOT, HEADMED, LITHO, BLUEBOX and other realized systems; NEOMYCIN, which attempted to reimplement MYCIN in a way closer to how clinicians actually work, and adds more structure to the data and rules, including frames, hierarchies, and meta-rules; RMYCIN, a direct re-implementation; and CENTAUR, a re-implementation of PUFF which explored the use of frames to enhance explanation.

CATEGORY TWO

Characterization of Givens

MYCIN uses static data in several forms. There are rules, context descriptions, a dictionary, tables and lists.

Rules are stored (in Lisp) as a set of properties on an atom representing the name of the rule. Properties include the premise and the action. These are templates that are instantiated into executable Lisp expressions. Different versions of MYCIN had 200 to 450 rules.

Context descriptions are type descriptions, and are described in more detail in Category Five. There were 10 to 13 contexts, with 60 to 200 clinical parameters.

The dictionary was used to aid in parsing, handling tasks such as synonym lookup, and correcting spelling errors.

Tables and lists are used to represent sets of values, such as known organisms, valid responses to culture sites, known characteristics of organisms, etc.

MYCIN uses dynamic data consisting mainly of a context tree (see below), but also including a record of rules used along with questions and answers in order to answer user queries and allow changes.

Characterization of Output

MYCIN produces a diagnosis of which organisms are present, and recommends a particular drug regimen. It can also provide justification for its reasoning.

Characterization of Data

The rules are reliable in that the overall system performance is of expert quality. The rules are incomplete in that they cover only a small domain of a clinicians' work.

Generic Tasks

Using Chandrasekaran's classification of generic tasks, MYCIN's diagnostic phase performs *classification*. Chandrasekaran mentions MYCIN and admits no other description, but one could view MYCIN's use of CFs as *hypothesis matching*. And certainly MYCIN's (later implementation of) therapy selection phase is *hypothesis matching*.

Theoretical Commitment

MYCIN's theoretical underpinnings include the use of production rules and backwards chaining, along with certainty factors (CF's). It claims that these mechanisms are easily understood by experts and can create useful systems. EMYCIN claims a broader usefulness for diagnosis problems.

Reality

There is no particular claim that MYCIN represents how clinicians reason at any detailed level. At a broad level, though, the steps of a typical consultation with MYCIN are modeled after a typical expert's: determination of whether a significant infection exists, what are the likely organisms, which drugs would be effective, and then which drugs would be most appropriate. The later NEOMYCIN did attempt to model human reasoning. It did this, in part, to provide more useful explanations.

CATEGORY THREE

Completeness

MYCIN was completely implemented, and in fact re-implemented.

Use

MYCIN has been used by real users, but only in evaluation. MYCIN was developed with the intention of solving a real-world problem. However, liability issues, among others, prevented its actual use.

Performance

There were three evaluation studies. The first two had experts evaluate MYCIN's performance in selected cases. It got approval ratings of 75%. The researchers were disappointed by this, but later found that this was a higher ranking than experts give each other.

The third evaluation was a blind study in which experts ranked performance by MYCIN against other experts and a student for a representative selection of meningitis cases. Clinical summaries were given to experts and to MYCIN, and their recommended treatments were recorded. The summaries and treatments were then compared by a wider group of experts. The student fared worse, and MYCIN fared better than the experts overall.

CATEGORY FOUR

Phases

MYCIN has two main phases: diagnosis and therapy selection.

Subfunctions

MYCIN's diagnosis phase is of interest because it attempted to generalize the representation of knowledge and reasoning in solving a classification problem.

The therapy selection phase was programmed in a much more problem specific way. Drugs are ordered by effectiveness for each organism, and then combinations are chosen until one succeeds.

Use of Simulation or Analysis

MYCIN does not simulate anything. It does do some analysis of drug dosages.

System/Control Implementation Architecture

MYCIN uses production rules with backward chaining as its main control architecture.

Since MYCIN deals with uncertainty, it must proceed down multiple paths. It uses an arbitrary cutoff (CF < .2) to trim the search tree.

CATEGORY FIVE

Characterization of Structure Knowledge

MYCIN uses the concept of context to structure knowledge. A context is a type that is like a restricted frame. There are contexts for patients, cultures, organisms, etc. They form a tree with the patient at the root. Rules referencing a particular object at some level in the tree may also reference the parents.

Each context has a set of associated characteristics or attributes, known as clinical parameters.

The contexts are used to control the use of rules, to guide instantiation of variables within rules. Clancey says that these contexts are really blackboards.

Rules reference object-attribute-value triples, where an object is of a contextual type, an attribute is one of the clinical parameters and a value is a valid value for the attribute.

Characterization of Process Knowledge

If process knowledge is dynamic knowledge, then in MYCIN, the dynamic knowledge represents its reasoning and the record of the steps it took.

Deep or Surface

MYCIN is relatively shallow in that no simulation occurs. However, the context structures and some of the built-in data structures represent some depth of knowledge.

CATEGORY SIX

Search Space

MYCIN searches through the space of possible patients, cultures, organisms, drugs, and regimens. All these objects have attributes, which in turn can have multiple values and certainty factors. The space is very large, since not only the attributes but the objects themselves can be multi-valued. For example, one patient can have a large number of cultures, each culture can have a large number of organisms, and so on.

The state is represented by the values in the context tree. Only the current state is represented.

Space Traversal

The state space is traversed by adding to or modifying the current context tree. This occurs when the user supplies input (direct facts) and when rule actions trigger.

Search Control Strategy

MYCIN uses backwards chaining of production rules, which is general technique with no claims of cognitive validity. The general strategy has been modified in several ways to get closer to an understandable reasoning, to achieve higher efficiency, and to avoid errors and nonsense.

Standard Search Strategies

The diagnosis phases uses depth first backwards chaining.

The selection of drugs uses Generate & Test.

Search Control Characterization

The basic control mechanism is backwards chaining with depth first search. However, modifications include:

- Use of LOOKAHEAD attribute of a clinical parameter to find rules that reference the parameter in their premise.

- Generalization of the sub-goal; MYCIN does not attempt to prove that a particular value is valid for an object-attribute; instead, it tries to find all valid values.
- MYCIN tries all relevant rules and then combines their CFs.
- MYCIN uses a CF cutoff strategy.
- MYCIN always tries to prove the MAINPROPs list associated with any instantiated object.
- MYCIN uses “antecedent” rules when a fact is asserted with certainty. These rules provides immediately known related facts.
- MYCIN has self-referential rules that can change the CF once a fact is known at all.
- MYCIN has mapping functions that allow it to reference contexts not on the current branch.
- MYCIN uses meta-rules.
- MYCIN prefers rules that assert with certainty.
- MYCIN casts out rules that have false premises (AND with any known false case.)

Sub-problems

MYCIN uses subgoals constantly, and assumes independence of context branches, but combines them using CFs.

Search Control Representation

As outlined above, there are many aspects to search control. Some are present in the context descriptions (MAINPROP). Some are in the rules (meta-rules). Some are implicit (CF cutoff).

Search Control Strength

MYCIN’s search control is generally domain independent. In several cases, though, the controls are triggered by the data; for example, the MAINPROPS attribute of a context tells MYCIN which objects to instantiate first independent of the rules.

CATEGORY SEVEN

Failure Method

At a global level, MYCIN incorporates explanation modes that help the user understand its reasoning. The user can identify which answers or rules led to the result. The user can change specific answers.

At a lower level, MYCIN is usually generating a set of possible results and using CF calculations to get a final result. So failure of a particular goal is part of the normal flow.

Uncertainty

MYCIN deals with uncertainty by using certainty factors (CFs).

MYCIN explicitly accepts user input that is uncertain. The user may append a parenthesized estimate of their certainty of any particular answer, and MYCIN carries that factor through its calculations.

Rule actions have CFs, as well.

And MYCIN reports its final results with attached CFs.

Management of Uncertainty

CFs nominally represent degree of belief in a hypotheses, but sometimes may be more akin to fuzziness. The CF value ranges from +1 representing certainty to –1 representing impossibility.

When a rule has several premises, the combined premise CF is formed by using the maximum of the OR factors and the minimum of the AND factors.

The result of a rule action with a CF is simply to multiply the rule's action's given CF by the premises' combined CF.

If a new result is asserted (C2) where that data (C1) already exists in the WS, the CFs are combined by the following rule:

C1	C2	Update
>0	>0	C1 + C2 - (C1*C2)
<0	<0	C1 + C2 + (C1 * C2)
otherwise		$(C1+C2)/(1-\min(\lvert C1\rvert,\lvert C2\rvert)$

Note that the result of this formula depends on the order in which C1 and C2 are asserted which can lead to inconsistent results.

When a rule resulting assertion is between –0.2 .. +0.2, MYCIN discards the assertion.

Also note that MYCIN does not allow OR as the top level function in a rule; instead several rules are created. The result of combining several results is different than the maximum of the premises.

Management of Time

There is no time-dependent data; or perhaps it is more accurate to say that any particular run of MYCIN is for a particular point in time. The VM system built on EMYCIN to deal with updating values over time.

CATEGORY EIGHT

Knowledge Representation Method

MYCIN represents static knowledge through rules and specific case knowledge through contexts.

Knowledge Representation Generality

Rules are specially coded Lisp structures. A separate mechanism exists that accepts natural language rules descriptions and runs through a verification step. TEIRESIAS improved this mechanism.

Knowledge Structuring

The context tree structure in MYCIN roughly corresponds to the domain and improves efficiency but is not a completely accurate model, since, for example, an organism may show up in several cultures.

CATEGORY NINE

Alternative Representations

MYCIN does not use multiple representations.

Alternative Solution Methods

None.

Optimization

There is no absolute measure for correctness in this domain. However, evaluation of MYCIN shows that it equals or out-performs experts in its domain.

Multiple Results

MYCIN explicitly deals with multiple results by using certainty factors throughout its diagnosis. It presents its resulting diagnoses with their certainties; it recommends a single treatment plan that attempts to treat all the diagnosed conditions. The treatment often includes multiple drugs.

CATEGORY TEN

Interaction

While MYCIN uses a textual interface, considerable effort was put into usability issues. In fact, the use of production rules was in part motivated by the need to explain its actions. The user can change a previous answer at any point; can question how a particular result was achieved or why a particular line of questioning is being pursued; and can reply unknown or provide a certainty factor for many answers.

Data collection

MYCIN works by asking questions to establish facts it needs to satisfy hypotheses; i.e., backward chaining. So the user interacts with the system as it works, and questions are pertinent to the problem at hand. Many changes to a basic depth first flow were made in order to ensure that the order of questioning made sense to the user.

Data format

Users interact with MYCIN by typing in a formalized subset of English. The system asks questions so as to limit the range of responses to a set that it can interpret. For example, no open ended questions are asked. It does some spelling corrections and can handle some synonyms. It does not use a strict parser, but instead concentrates on keywords and context.

Rules are stored in Lisp in the original system; they can be viewed and added in natural language. See TEIRESIAS for later changes.

Acquisition

MYCIN has a knowledge acquisition module which produces rules in a stylized subset of Lisp. The original mechanisms was replaced by TEIRESIAS.

Learning

MYCIN does not learn. Rule changes are manually made by expert users of the system, or by programmers/Knowledge engineers.

Explanation

MYCIN records questions and answers and rule chains in order to answer questions about why it asks questions and how it arrives at results.

CATEGORY ELEVEN

Strengths

- Performed as well as experts.
- Led to a whole generation of expert systems.
- Dealt with uncertainty in a useful way.
- Explicitly dealt with usability issues, according them great importance from design on.
- Provided visibility into its reasoning.
- Structured data in a useful way.
- Attempted to really solve an important problem.

Weaknesses

- Ad hoc mechanism for uncertainty is inconsistent.
- Data structures and rule control too specific.
- Explanation mechanism not always helpful.
- Didn't give user enough control.
- Inability to update over time.

Other

XXX