GENERATION AND MAINTENANCE OF ILIAD™ MEDICAL KNOWLEDGE IN A HYPERCARD™ ENVIRONMENT

Dean Sorenson, Omar Bouhaddou, Wenshao Wang,
Gerald Canfield, Lee-Shing Fu, & Homer R. Warner

Department of Medical Informatics
University of Utah

Abstract

A comprehensive system has been constructed to systematically and efficiently 1) design, translate, compile, and test new knowledge frames, and 2) update and facilitate access to text frames, associated literature citations, and statistical information. The availability of the knowledge building tools in a unified workstation environment has synergistically accelerated the process of building, updating, and maintaining medical knowledge bases.

Introduction

An unusual feature of the HELP system [1] as a hospital information system is its significant decision support component. This decision support component (i.e., the knowledge base) is comprised of about 2000 decision frames. Various versions of these knowledge frames have been used on a daily basis for more than 10 years as an integral part of the medical decision making process at LDS hospital in Salt Lake City.

In addition to the knowledge base associated with the HELP system (which is maintained on a mainframe computer), an independent knowledge base has more recently been developed and implemented on Macintosh computers at the U. of Utah Dept. of Medical Informatics. This knowledge base was developed specifically for ILIAD [2] and is comprised of approximately 185 frames representing relationships used for diagnosing diseases in four medical specialties (cardiovascular, pulmonary, gastrointestinal, and endocrine diseases). About 60% of these frames are probabilistic (i.e., use Baysian inferencing) in nature. An average probability frame has about 20 findings, each with an associated sensitivity (probability of the finding being present in the disease population) and 1-specificity (probability of the finding being present in the non-disease population). These numbers are obtained from the patient data base, the literature, or from expert estimates.

Two additional subknowlege bases in nephrology and hematology will be completed by August 88. Several others are under development.

The construction of knowledge frames has now been formalized and partially semiautomated. The process involved is the subject of this paper.

Knowledge Frame Construction

Knowledge frames are constructed through the process of knowledge engineering. This process involves:

- the identification of the most appropriate subset of findings necessary and sufficient to make a diagnosis or other medical decision
- a decision as to whether the frames should be probabilistic or deterministic
- identification of non-independence
- a decision as to whether the frames should be all individual items or whether there should be clusters [3] of findings (clusters are a strategem designed to abrogate lack of independence and to group symptoms with common pathophysiological processes or mechanisms, thereby increasing the educational value of the frames)
- assignment of probabilities to the individual items
- testing item probabilities relative to
- testing the medical logic of the frame relative to other diseases which share some of the findings of that frame

0195-4210/88/0000/0366$01.00

The knowledge engineering process has been recently formalized at the University of Utah. This process involves regular meetings between domain experts, knowledge engineers, data base experts, library scientists, computer scientists, and graduate students in medical informatics. A dynamic record of the proceedings are maintained using a computer application called KESS previously described at the SCAMC87 meetings [4].

Integration of the knowledge frames into a specific educational or clinical environment requires input from additional experts trained in statistics, sociology, and the cognitive sciences as well as educators and clinical supervisers. This information is assimilated during weekly meetings.

The knowledge frames can be updated at any time as deficiencies are discovered in the logic, additional findings are added, better estimates of the probabilities are obtained, or findings are "elsed" or combined into clusters. However, adding findings to a given frame can require changes in the data dictionary. Since the ILIAD data dictionary has a hierarchical structure (necessary to facilitate the sophisticated inferencing capabilities of ILIAD), the addition of new terms must adhere to certain rules. Also, when new frames are built, the user must be able to access the existing data dictionary. In addition, it is desirable in this regard, to provide access for computer-naive medical practitioners as well as veteran users.

To facilitate these logistic considerations, an integrated computer system has been developed for the generation and maintenance of computerized medical knowledge. This system is comprised of the following components (Figure 1): dictionary manager, translater, compiler, data base, and testing utilities. Each of these components are further described below:

The Dictionary Manager

The HELP system data dictionary (PTXT) is a hierarchically arranged dictionary of medical terminology (e.g., all SNOMED terms) that uses numerical codes to represent the hierarchy. The ILIAD data dictionary is a subset of the HELP system data dictionary.

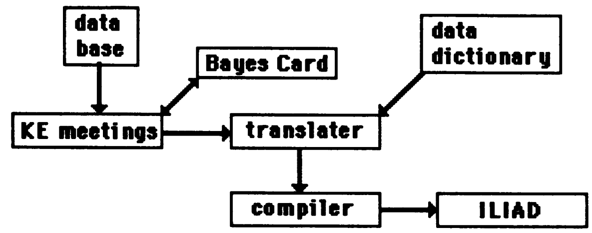

Figure 1. Overall design and flow of information between the various components of the ILIAD/HyperCard knowledge frame generation and maintenance system.

The smaller size of this dictionary (presently only about 1500 terms) has facilitated a substantial restructuring of the hierarchy to optimize it's use in the ILIAD environment. This restructuring included elimination of redundancy, adding terminology which makes each item stand alone (necessary in the ILIAD environment where isolated questions are presented to the student), elimination of hierarchies that are not useful for inferencing, addition of structures that are useful for inferencing (e.g., high-level nodes with terms such as abnormal or increasing), and the storage of frequency of occurrence with corresponding dictionary terms. The storing of probabilities (i.e., total probability of a finding across all diseases) within the data dictionary, in turn, facilitates additional functionality such as the use of partial information (see below), consistency checking (e.g., is the sum of the probabilities of all the children in a mutually exclusive set less than or equal to that of the parent) and transportability of the knowledge base to other locations.

The dictionary manager has the following functions: editing (i.e.., adding, deleting, and changing dictionary terms), inferencing, and consistency checking. A set of data manipulation algorithms have been constructed to facilitate 1) transfer of data from application programs, 2) the location of appropriate storage space, 3) storage of data items, 4) deletion of data items, and 5) the ability to change data items. A keyword retrieval and template query system has been designed to maintain and make optimal use of the existing structures in the expert system and, at the same time, to provide a convenient user interface.

The Translater

The translater is a computer program which converts a nearly free-text version of a knowledge frame to a representation which can be inputted to the compiler (see Figure 2). This program is comprised of a *lexical analyzer*, parser, dictionary retrieval component, and precompiler. The lexical analyzer breaks up a nearly free-text document into its component parts: title, type, findings, probabilities, and logical statements.

The *parser* is comprised of a retrieval module, logic editor, and precompiler. The retrieval module deletes meaningless words from each statement and then uses the meaningful terms to find the appropriate hierarchical data dictionary terms. This process involves the use of a morphology algorithm which sequentially removes letters from the end of each word and matches them to a suffix list. The resulting word stems are then compared to a list of synonyms and words in the data dictionary. The most appropriate term or terms from the data dictionary are then presented to the user for approval. Rules are being developed to disambiguate the appropriate terms when multiple hits are made. The user also has the option of retrieving the entire list if desired; then choosing the most appropriate term.

The parser has a logical editor component which puts the statements in the frame in the appropriate order based on information content. The importance of a given term in a probability frame is a function of the ratio of the sensitivity to (1-specificity). When statements are not independent, they must be grouped by OR or ELSE statements. In an interpretation frame, the parenthesis must be automatically placed appropriately so that the AND and OR statements make sense. Also, the appropriate negative logic is automatically produced from the positive logic.

The parser generates the following output: 1) a code which is understandable by the computer programmer which involves search, arithmetic, and probability statements concerning existence, value, and time variables, 2) the production of a numerical code which can be inputted to the compiler.

The user interface is menudriven and follows the usual Macintosh conventions. The construction of new frames is facilitated by an interactive environment with examples and dialogue boxes which inform the user what he/she can or cannot do. The usual functions of cut, copy, paste, save, print, etc, are supported. When the text version is completed, the user directs the computer to parse the frame to generate a pre-compiled version of the frame (Figure 2). The frame is subsequently compiled and can then be immediately tested in the ILIAD environment.

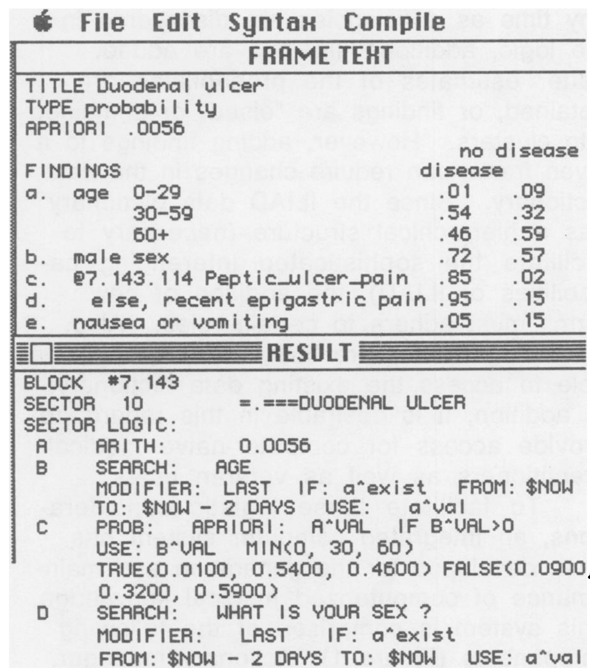

Figure 2. Free text version of a knowledge frame (in proper format) and the corresponding precompiled version generated by the translater.

The Compiler

The Macintosh compiler is a "C" program that converts the precompiler version of the frames to machine-readable code (pcode). The following data files are generated: frame file, item file, code file, rank file, dictionary file, and keyword file. The frame file contains the compiled version of the knowledge frames and the dictionary file contains the ILIAD data dictionary hierarchical codes. The item file contains a list of the items in the frames and the links between items in the same frame. The keyword file allows dictionary items to be accessed by the user during runtime. The code file contains the links between a given finding and all the frames that use it. The rank file orders items by information content (i.e., a function of sensitivity/(1-specificity)) and cost. Also, the system uses this file to keep track of what questions have been asked during runtime.

The compiler also generates a truth table which allows the status of every item entered while using ILIAD to be tracked. The status is determined by the existence, value, time, and time limit of each item entered.

Hypercard

Hypercard™ is a Macintosh application with likens each record to a "card", with sets of cards being contained within "stacks" (analogous to files in a file drawer). Graphics and text can be freely mixed on cards. Cards and stacks are accessed by clicking a "button" or selecting a menu item on the "Home card" with the "mouse". Each stack has a top directory card which facilitates access to the other cards in the stack. The previously described parts of the ILIAD knowledge base generation and maintenance system (except for the KESS system which is in Hypercard™) are "C" programs which are accessed via a Hypercard™ interface. The data base and evaluation tools for the system are all in Hypercard™.

The Data Base

The data base is wholely contained within Hypercard™. The data base is comprised of the following stacks: text frames, findings, statistics, and literature. The text frames component contains each of the frames in scrollable fields on separate cards. These cards are accessed by selecting a title in a scrollable directory and clicking a button with the mouse. Clusters can be accessed from each frame by the same hypertext mechanism. Clicking another button returns the user to the original frame card. The relevent contents of each probability text frame can be transferred to a Baye's testing card (see below) by clicking a button. Another option takes the user to a card in the statistics stack delineating the source of the statistics used in the frame.

The first card of the findings stack contains a scrollable list of all of the findings contained in the ILIAD frames. Clicking the name of a finding with the mouse brings up a list of all of the frames which use that particular finding.

The literature stack contains a scrollable list of keywords in the directory card. Buttons and menu selections then take the user to the desired citations or retrieve a list of citations according to a set Boolean criteria.

The Baye's Card

In the case of probabilistic frames, the symptoms of a disease are assigned associated numerical values (i.e., sensitivities and specificities). ILIAD uses these numerical weights in a Bayesian framework to calculate a posterior probability of a given disease. To help the knowledge engineering team to intuitively comprehend the impact of each symptom's participation in the final posterior probability of a disease, the "Baye's card approach" was developed. This HyperCard™ stack lets the user "import" a free-text format of an ILIAD knowledge frame and simulate a status (e.g., present, absent, or unknown) for each manifestation of the illness. Another button allows the knowledge engineer to calculate the likelihood of the disease for the set of simulated patient observations. If the posterior probability of the disease is not in the range of that expected by an expert, given the patient case, then the appropriate probability estimates are adjusted until a satisfactory behavior evolves. Medical experts participating in the knowledge engineering sessions have found this tool very useful since it allows them to focus on the relative contribution of an item, rather than on individual numerical weights. This is a method of adjusting the overall reliability of the decision model in a manner analogous to tuning a radio without having to be concerned with the individual frequencies.

An additional option of the Baye's card, found to be useful in terms of evaluation of the model, is a stochastic simulation mode. In this mode, the presence of a disease is assumed, findings are randomly generated, and the corresponding posterior probabilities are calculated. This process is repeated many times and the results plotted. When most of the results fall close to 1.0, it may be assumed that the disease is well described by the corresponding set of findings and assigned weights. However, there are cases where the interpretation is not straightforward, because of a need to use nonspecific findings (i.e., findings common to many diseases) or when there are few known findings associated with a disease.

A version of the Baye's card has also been constructed for the testing and evaluation of interpretation type knowledge frames. This version is based on a newly developed algorithm for ranking non-probabilistic findings [2].

Discussion

Our previous experience with the knowledge engineering process has shown that the process can be incredibly inefficient and prone to errors if done on a piecemeal basis. This is because of the multidisciplinary nature of the process where a number of individuals from different departments, with different time requirements, are contributing to the process. The frame translation process is especially prone to errors, which are sometimes very difficult to track down, since the process is tedious and difficult to standardize, and is likely to be done by persons who wouldn't recognize certain medical nuances. These considerations prompted the design of a semi-automated, modularized, yet integrated, system to facilitate the process. By automating this process, the turnaround time is greatly shortened, since a frame can be designed, translated, compiled, and tested entirely during a single session while all of the pertinent information fresh in the minds of the designers.

In order to ensure optimal design of the data dictionary, we found it necessary to establish formal fules for the structure of the data dictionary, and to assign a data dictionary manager for this purpose. Likewise, a data base manager was assigned the responsibility of maintaining the data base and making all necessary changes.

The establishment of a central data base greatly reduces inefficiency, duplication, and loss of time and information. For example, it is frequently necessary to refer to previous frames when building new frames so as to prevent duplication of effort and to facilitate consistency in the assigned probabilities. In terms of documentation, the statistics and literature parts of the data base are essential.

We have found that the testing of the probability frames in the Bayes card environment invariable improves the performance and reliability of incipient frames. It is substantially more efficient at this stage than the succeeding one (in the ILIAD environment) where there are the additional complexities of multiple frames present.

References

[1] Pryor, TA, Gardner RM, Clayton PD & Warner HR. The HELP system. J. Medical Systems 7:87-102 (1983).

[2] Warner HR, Haug P, Lincoln M, Warner Jr. H, Sorenson, D, Fan C, Bouhaddou O, & Williamson J. ILIAD: A computer aided program for teaching differential diagnosis to medical student. In press (SCAMC 1988).

[3] Lincoln M, Turner C, Hesse B, Warner HR, & Miller R. Discovering clustered disease findings: prospects for enhancing expert

[4] Ben Said M, Dougherty N, Anderson C, Altman SJ, Bouhaddou O, Warner HR. KESS: Knowledge engineering support system. Proceed. 11th Sympos. Computer Applications in Medicine 56-59 (1987).