Automatic Construction of Medical Cases for Training and Testing Using the Knowledge Base of an Existing Decision Support System

Sandra Carberry
Dept. of Computer Science
University of Delaware
Newark, DE 19716

John Clarke M.D.
Dept. of Surgery
Medical College of Pennsylvania
Philadelphia, PA 19129

Abigail Gertner
Dept. of Computer Science
University of Pennsylvania
Philadelphia, PA 19104

Introduction

Early AIM research recognized the potential for using the knowledge base of a medical decision support system for training and testing students. Clancey[1] took on the entire task of restructuring both MYCIN's knowledge base and its reasoning methods to do this. Our goal is simpler but potentially more widely applicable: to use existing knowledge bases to automatically generate realistic cases that could be used for instructional purposes by a training module or for computer-based recertification exams by a quality assurance module. Among other benefits, this would eliminate the need to collect and pre-store a library of cases, and would greatly reduce the likelihood that a selected case replicates one used previously, and the output would be available to a variety of application programs. For example, an interactive program might want to make use of skeletal cases initially, with the case generation system fleshing out further details of the case as the student or examinee responds with bedside questions, test orders, and therapeutic actions. In this way, the case could be adjusted based on the user's degree of success in diagnosing the case thus far, with the case becoming more complex if the user is demonstrating expertise but becoming more straightforward if the user is having difficulty selecting appropriate steps. Such adjustment reflects standard practice in medical oral exams, where the examining physician adjusts details of a case to elicit further reasoning by a student or to illustrate ramifications of a student's error.

Our long-term goal is to develop a system for automatic incremental case generation that takes advantage of the extensive knowledge already encoded in the knowledge bases of existing decision support systems. As a first step, we have focused on generating complete cases of varying degrees of difficulty based on a single diagnosis. (A complete case is a complete set of symptoms, findings, results of diagnostic tests and actions, etc.) Although other researchers have investigated the generation of simulated patient cases, their knowledge bases and/or kinds of cases have been different. Most of these simulations, such as those described in [1, 3, 5], rely on knowledge bases that detail the manifestations associated with each disease along with their frequencies of occurrence; cases are generated by randomly selecting manifestations for inclusion in the case, with the manifestation's probability of inclusion determined by its frequency of occurrence. While these systems have produced good results, these methods of generation do not permit flexibility in generating cases of varying degrees of difficulty or facilitate incremental case generation. In contrast, we have been concerned with a decision support system whose knowledge base consists of rules for developing diagnostic and therapeutic plans for managing patient care. Our method for generating cases relies on chaining backwards through selected rules, and we believe that it can be extended to generate cases that are dynamically adapted to the student's individual needs.

Eliot and Woolf[4] developed a somewhat different approach to producing simulated cases. Their system works in the domain of cardiac resuscitation, and a simulation consists of a series of transitions between states of cardiac arrest or arrhythmias. While their system uses the probabilities of state transitions to guide the simulation, it also includes a mechanism that biases the system to enter states that provide an opportunity to teach high priority topics. However, the kind of case they produce differs from ours in that the generated case is a series of patient state transitions rather than a set of consistent symptoms and findings that warrant a particular diagnosis.

This paper describes the principles underlying our implemented system, presents an example of a case generated by the system, and outlines our current work on extending the system to generate cases of varying degrees of difficulty.

Case generation

Our case generation system starts with an intended overall diagnosis, such as a tension pneumothorax, selected from a list of possible diagnoses. The case produced by the system consists of a set of initial signs and symptoms, along with symptoms that can be elicited through appropriate bedside questions and results of appropriate diagnostic procedures and tests. For case generation, signs, symptoms, and test results can be seen as falling into four overlapping classes:

- C1: those suggestive of the specified diagnosis
- C2: those required for the specified diagnosis
- C3: those typical of the specified diagnosis
- C4: those unrelated to the specified diagnosis.

These four classes play different roles in automatic case generation. Not only must a good case include symptoms and/or test results that lead the physician to conclude the specified diagnosis [C2], but it must also include features that suggest considering that diagnosis [C1]. For example, while a positive finding from a needle aspiration of the chest for pressure will produce a diagnosis of tension pneumothorax, the case must also include symptoms such as a chest wound, shock, distended neck veins, and decreased breath sounds that would lead a physician to consider a tension pneumothorax and perform the needle aspiration. In addition, although there are many ways that a conclusion might be drawn, it is important that the case generally reflect a representative situation [C3]. For example, while one might conclude tension pneumothorax from either a positive needle aspiration of the chest for pressure or from an xray showing a tension pneumothorax, typically physicians diagnose this condition by doing a needle aspiration of the chest. Thus a typical case should rely on the needle aspiration instead of the chest xray.

We are seeking to identify principles that apply generally to case generation. However, to ground and implement our ideas, we have applied them to trauma care and have developed a system that generates cases by reasoning on the knowledge base of TraumAID[6], a decision support system for addressing the initial definitive management of multiple trauma. TraumAID's three part knowledge base is encoded declaratively and contains extensive medical knowledge related to trauma care. It consists of 1) evidential rules that derive further conclusions from existing evidence, 2) goal setting rules that post goals based on accumulated evidence, and 3) mapping rules that specify alternative procedures for satisfying a goal and ordered sets of actions comprising procedures. Initial evaluation and validation studies indicate that TraumAID produces high-quality diagnostic and therapeutic plans for managing patient care in both simple and complex trauma cases.

Our problem was to identify how the information comprising classes [C1]—[C4] needed for case generation could be extracted from such a knowledge base. We found that this could be accomplished by chaining on TraumAID's evidential and goal setting rules and by relating TraumAID's mapping rules to its knowledge base about possible test results. By starting with the TraumAID rules that post a goal of ruling out the specified diagnosis (such as the rules that post the goal of ruling out a tension pneumothorax), and reasoning backwards on the goal setting and evidential rules, our system identifies factors that suggest the specified diagnosis. By reasoning backward from the TraumAID rules whose conclusion is the specified diagnosis, our system identifies alternative sets of symptoms and test results that will produce the desired diagnosis. The goal-procedure mappings specify the procedures that a physician should follow in pursuing a diagnostic or therapeutic goal. By examining these mappings and accessing TraumAID's knowledge about the possible results of different procedures and tests, our system is able to select test results that are representative of typical trauma emergencies. For example, the TraumAID goal-procedure mapping for pursuing the goal of ruling out a tension pneumothorax specifies doing a needle aspiration of the chest for pressure. Thus although the goal setting and evidential rules indicate that tension pneumothorax can be concluded either from a chest xray or from a needle aspiration of the chest for pressure, the latter will be selected in generating a typical case. (The system still has the ability to generate atypical cases, if desired.)

Thus our method for identifying the requisite conditions for a diagnosis of XD is the following:

1. Form a set R_s of the rules whose conclusion is Rule-out_XD, where XD is the desired diagnosis.
2. For each rule in set R_s, chain backwards to identify alternative sets of primitive elements[1] that must be present or absent for a diagnosis of XD to be suggested. Form set C_s as this set of alternatives.
3. Form a set R_c of rules whose conclusion is XD, where XD is the intended diagnosis.
4. For each rule in set R_c, chain backwards to identify alternative sets of primitive elements that must be

[1] Primitive elements are those that represent symptoms, findings, or test results, such as *distended neck veins* or *positive x-ray result for tension pneumothorax*; non-primitive elements are those that can be further decomposed into more basic elements, such as a finding of *clinical signs of intra-abdominal injury*.

present or absent for a conclusion of XD to be drawn. Form set C_c as this set of alternatives.

5. Where a goal of ruling out a particular diagnosis is posted (for example, RO_Tension_Pneumothorax), access its goal-procedure mapping to identify the preferred procedure for pursuing this diagnosis. Chain through the rules to find appropriate primitive tests or diagnostic procedures, and then identify the possible results of these procedures that are part of an alternative in C_c and mark these alternatives as typical.
6. Each combination of an alternative from set C_s and a consistent alternative from set C_c provides the facets of a generated case. Typical cases are those that contain an alternative from set C_c that has been marked as typical.
7. Randomly select from among the typical cases. (If an unusual situation is desired, then select from among the cases that have not been marked as typical.)

Example of Case Construction

To illustrate our system with a simple example, consider generating a case with a diagnosis of a left tension pneumothorax. Chaining backward from the only rule that posts a goal of ruling out tension pneumothorax leads to three alternative rules, all of which share the following elements:

True	Shock
True	Distended_Neck_Veins
True	Decreased_Breath_Sounds (Side=LEFT)
False/Unknown	Tension_Pneumothorax_Unlikely (Side=LEFT)

The three alternatives respectively include one of the following additional sets of elements:

Alternative-1a:

True	Wound (Location=CHEST,Side=LEFT)
False/Unknown	Open_Sucking_Chest_Wound (Side=LEFT)

Alternative-1b:

True	Wound (Location=CHEST, Type=GUNSHOT, Side=RIGHT)

Alternative-1c:

True	Wound (Location=ABDOMEN)
False	Distended_Abdomen

Tension_Pneumothorax_Unlikely is not primitive, so it requires further chaining, which produces the following set of three alternatives:

Alternative-2a:

True	Needle_Aspiration_Chest_For_Pressure (Side=LEFT,Test-Result=NEGATIVE)

Alternative-2b:

False	X_Ray_Tension_Pneumothorax (Side=LEFT,Test-Result=POSITIVE)

Alternative-2c:

True	Wound(Location=EPIGASTRIC)
True	Tension_Pneumothorax(Side=RIGHT)
True	Single_Wound

Since Tension_Pneumothorax_Unlikely must be False or Unknown, we must ensure that none of the above alternatives are present in the constructed case. This indicates that our selected set of conditions must not include a negative result for needle aspiration of the chest for pressure, must not include a left chest xray that does not produce a finding of tension pneumothorax, and must not include all three of single wound, epigastric wound location, and a right tension pneumothorax.

Next we must chain on rules whose conclusion is Tension_Pneumothorax in order to identify the conditions that must be satisfied to make the desired diagnosis. This produces the following alternatives:

Alternative-3a:

False	Needle_Aspiration_Chest_For_Pressure (Side=LEFT,Test-Result=NEGATIVE)

Alternative-3b:

True	X_Ray_Tension_Pneumothorax (Side=LEFT,Test-Result=POSITIVE)

This produces a set of alternatives as shown in Figure 1.

The facets for a possible generated case are identified by traversing one of the paths in the network shown in Figure 1, setting patient characteristics and test results as indicated. Note that the upper part of the figure presents the characteristics (preceded by asterisks) that are necessary for the physician to *consider* a diagnosis of a left tension pneumothorax, while the bottom part of the figure presents the characteristics (preceded by a pound sign) that are necessary for a conclusion of a left tension pneumothorax. For example, Figure 2 summarizes two cases that might be generated.

The goal-procedure mapping for ruling out a tension pneumothorax specifies doing a needle aspiration of the chest for pressure. Therefore, alternatives selected from set C_c (alternatives at the bottom of the

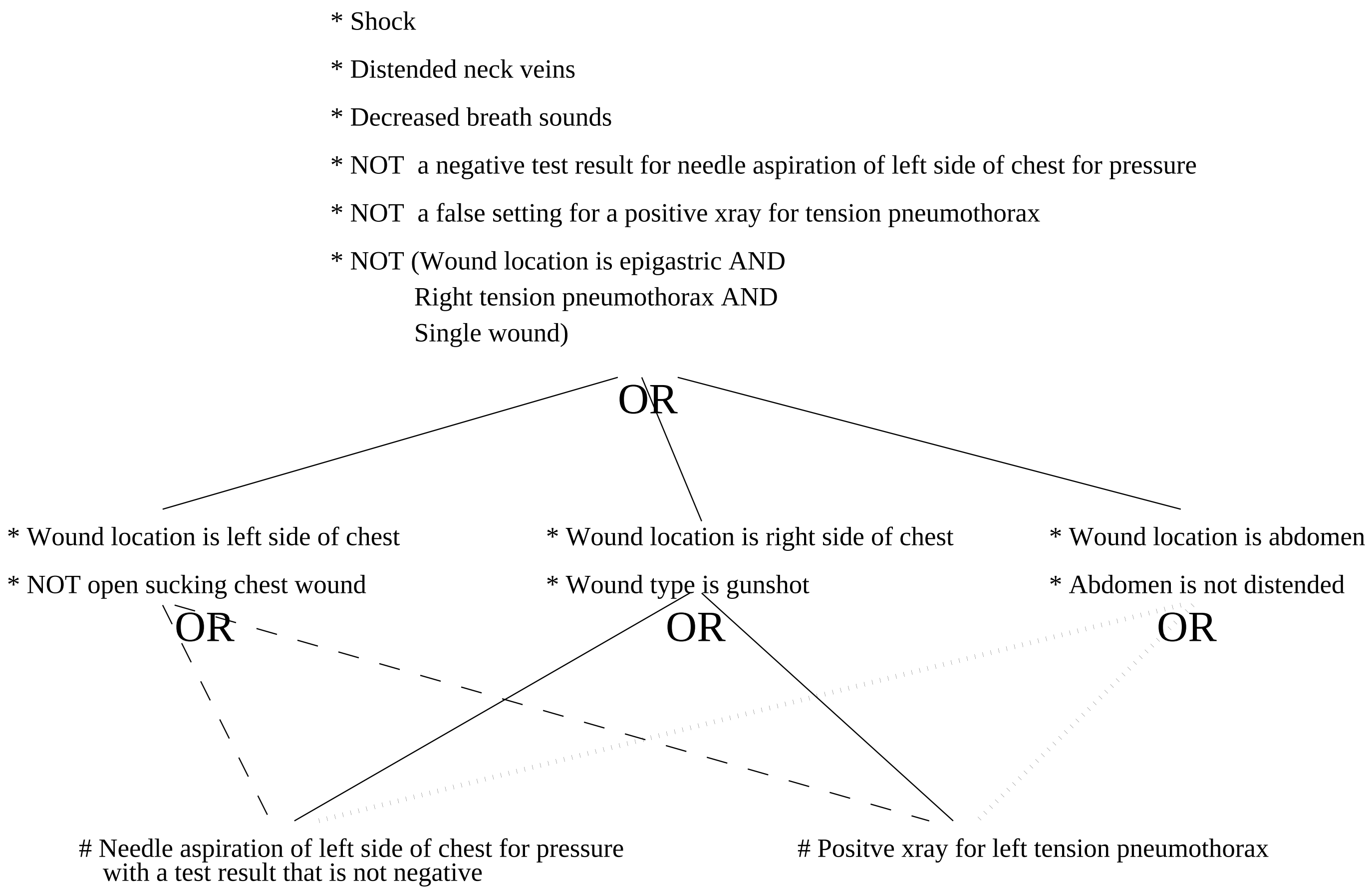

Figure 1: Essential Conditions for a Tension Pneumothorax Case

---

CASE-1:
Patient Characteristics:
∗ Wound location is left side of chest
∗ Wound is not an open sucking chest wound
∗ Shock
∗ Distended neck veins
∗ Decreased breath sounds
Test Results:
Positive xray for left tension pneumothorax

CASE-2:
Patient Characteristics:
* Wound location is abdomen
* Abdomen is not distended
* Shock
* Distended neck veins
* Decreased breath sounds
Test Results:
Needle aspiration of left side of chest for pressure
with a test result that is not negative

Key:
* Suggests considering diagnosis
# Required for concluding diagnosis

Figure 2: Two Sample Cases for Tension Pneumothorax

---

network in Figure 1) that contain a result of doing a needle aspiration are marked as *typical*. Consequently, CASE-2 is identified as a typical case while CASE-1 is not.

Further Work on Case Generation

We are now extending our system to generate cases of varying degrees of complexity. We have identified a number of factors that appear to correlate with the estimated difficulty of a case. Some of these factors (such as the presence of multiple injuries) are specific to the domain of trauma, but other factors (such as the number of alternative diagnoses that must be ruled out and how quickly each can be eliminated, whether test results are definitive or introduce ambiguity into the case, etc.) are domain-independent.

We have identified how our system can use the TraumAID knowledge base to estimate the complexity of a case as it is being constructed and to select features that produce a case within a predetermined range of difficulty. For example, one significant factor is the number of additional diagnoses that must be considered. A conclusion of tension pneumothorax along with shock despite chest decompression leads to the need to rule out a pericardial tamponade. If Continued_Neck_Vein_Distention is set to False (ie., the neck veins are no longer distended after the needle aspiration has been performed), then one set of findings that would produce a diagnosis of pericardial tamponade is eliminated. Others can be blocked in a similar manner. Since the TraumAID rules specify the requisite conditions for drawing a particular conclusion, they can be used to establish patient characteristics that block a conclusion and thus terminate an inference chain. We hypothesize that the number of potential diagnoses, the extent to which each must be investigated by the physician, and the length of each inference chain affect the difficulty of a generated case.

A very different factor that influences the difficulty of managing a case effectively is the presence of alternative procedures. For example, suppose that the procedure of choice for addressing two goals differs from the procedure that should be used for either goal by itself. Then a case in which both goals occur would conceivably present additional complexity in that the physician must be sensitive to the multiple goals when developing a management plan. Using TraumAID's goal-procedure mapping rules, our system will be able to alter case difficulty by enabling or blocking such interacting goals.

We are now extending our implementation so that the system will produce cases with a specified level of complexity. Eventually, we intend to produce a system that generates cases incrementally and can dynamically adjust the difficulty of a case based on the user's success in handling it.

Conclusion

This paper has briefly described a system for automatic generation of medical cases from the knowledge base of an existing decision-support system. Our work suggests that automatic case generation may well be feasible. If so, it makes good on the promise that knowledge encoded in medical knowledge bases could be used for the training and testing of medical personnel. In the future, combining realistic case generation with realistic human figure animation may provide even more, in the form of "life-like" virtual casualties for the immersive VR training systems of the future.

Acknowledgements

This work was begun while the first author was on sabbatical at the University of Pennsylvania and was supported by NLM under grant R01 LM05217. The research and paper have benefitted greatly from the comments and suggestions of Bonnie Webber.

References

[1] Homer Chin and Gregory Cooper. Case-based tutoring from a medical knowledge base. *Computer methods and Programs in Biomedicine*, 30:185–198, 1989.

[2] William J. Clancey. *Knowledge-Based Tutoring: The GUIDON Program.* MIT Press, Cambridge, MA, 1987.

[3] Robert Cundick, Charles Turner, Michael Lincoln, James Buchanan, Curtis Anderson, Homer Warner, and Omar Bouhaddou. Iliad as a patient case simulator to teach medical problem solving. In *Proceedings of SCAMC*, pages 902–906, 1989.

[4] Chris Eliot and Beverly Woolf. An adaptive student centered curriculum for an intelligent training system. *User Modeling and User-Adapted Interaction*, 5(1):67–86, 1995.

[5] R.C. Parker and R.A. Miller. Creation of realistic appearing simulated patient cases using internist-1/qmr knowledge base and interrelationship properties of manifestations. *Methods of Information in Medicine*, 28(4):346–351, 1989.

[6] Bonnie L. Webber, Ron Rymon, and John R. Clarke. Flexible support for trauma management through goal-directed reasoning and planning. *Artificial Intelligence in Medicine*, 4:145–163, 1992.