Augmented Transition Networks as a Representation for Knowledge-Based History-Taking Systems

Alex D. Poon, Kevin B. Johnson, Lawrence M. Fagan

Section on Medical Informatics
Stanford University School of Medicine
Medical School Office Building (MSOB X215)
Stanford, CA 94305-5479

ABSTRACT

Numerous history-taking systems have been built to automate the medical history-taking process. These systems differ in their control methods, input and output modalities, and kinds of questions asked. Thus, there has emerged no standard way of representing interviewing knowledge—the expert knowledge used to govern the sequence of questions asked in an interview. This paper discusses how we use an augmented transition network (ATN) to represent the knowledge of a speech-driven automated history-taking program, Q-MED, and how, more generally, ATNs could be used as a representation for any knowledge-based history-taking system. We identify three characteristics of ATNs that facilitate the use of ATNs in interviewing systems: explicitness, hierarchical structure, and generality.

INTRODUCTION

The goal of an automated history-taking system is to allow a patient, from a large list of symptoms, to specify which symptoms are present and which are absent. One way to accomplish this goal is to have patients read through a long list of symptoms sequentially, and, for each symptom, to select "yes" if the symptom is present, and "no" if it is not [1]. Of course, such a procedure can be painstakingly slow, especially if there are many symptoms in the list. Some automated history-taking programs employ branching techniques to mimic the history-taking interviews of health-care professionals [2-6]. Such programs reduce the number of questions asked of the patient by using knowledge about the medical domain to skip irrelevant questions. Still others use statistically based control methods, such as a sequential Bayesian approach, to determine the ordering and selection of pertinent questions [7].

Automated history-taking programs differ not only by the control methods used to select questions, but also by the input modalities, output modalities, and kinds of questions asked. Keyboards [3, 4, 6], light pens [2, 8], touch-tone telephones [5], and speech recognizers [9] have all been used as input devices to history-taking programs. Graphics [2, 8], text [3, 4, 6], color film [8], and voice [5] have been used as output modes. Most systems ask yes–no and multiple-choice questions [2-6, 8-10]; some use only yes–no questions [1, 7]; and at least one system employs open-ended questions [9].

With all the different ways that automated history-taking programs have been implemented, there has emerged no standard way of representing interviewing knowledge—the expert knowledge used to govern the sequence of questions asked in an interview. In this paper, we will discuss our use of an augmented transition network (ATN) to represent the knowledge of a speech-driven automated history-taking program, Q-MED. We investigate how, in general, ATNs can be used for developing any knowledge-based history-taking system.

BACKGROUND

Most computer-based history-taking systems represent interviewing knowledge as a set of questions with contingencies for selecting subsequent questions in the interview. The Automated Medical History (AMH) by Mayne, for example, specifies with each question the image to be displayed on the screen, the set of legal responses, the storage for the user's response, and a set of IF statements that determine the branching logic. All questions are yes–no or multiple choice. [8]

Converse, by Bloom, uses a similar representation. In this system, a frame is the basic unit of the interview; a frame consists of the text of a question, the storage for the response, and a set of IF statements. Frames can be grouped into sections that can be executed as a unit. A typical IF statement is "IF THE RESPONSE CONTAINS 1, DO 536," where "1" is a multiple-choice selection and "536" is the next frame or section to invoke. IF statements may refer to the response to any frame, not just to that for the current frame. Responses can be free text, yes or no, multiple-choice selections, or entries in a table [10].

Warner's representation scheme for history taking differs greatly from the ones already mentioned. Warner uses a matrix in which each row in the matrix represents a diagnosis, and each column represents a yes–no question. The cells of the matrix represent the likelihood that a patient with the corresponding disease would answer "yes" to the corresponding question. The program then uses a sequential Bayesian approach to select the most useful question to ask next for determining a diagnosis [7].

HOW Q-MED USES AN ATN TO REPRESENT INTERVIEWING KNOWLEDGE

0195-4210/92/$5.00 ©1993 AMIA, Inc.

Our speech-driven interviewing system, Q-MED, uses the ATN formalism to represent the interviewing knowledge of a system for interviewing patients who have back pain. ATNs were originally developed for natural language processing. We refer the reader elsewhere for a description of the ATN formalism [14,15]. We divide our ATN for back pain into two levels, separating the knowledge of the overall structure of the interview from the knowledge of the ordering of specific questions. On the higher, more general level, *nodes* represent general domain concepts, such as the location of the pain or the activities that exacerbate the pain. Each such concept node is associated with a lower-level sub-ATN whose nodes represent specific questions related to that concept. Figure 1 shows nodes from the ATN of the back-pain version of Q-MED.

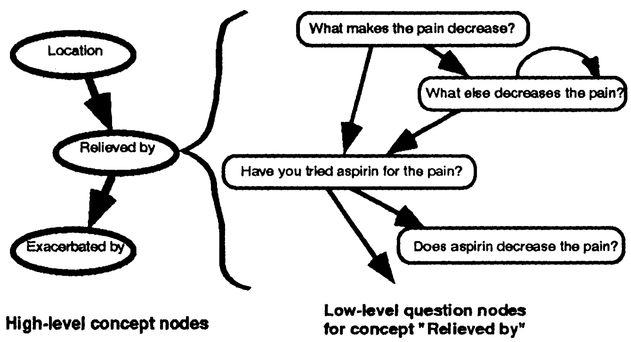

Figure 1. Nodes showing the two-level structure of the back-pain ATN. The ovals on the left represent concept nodes, while the rounded rectangles on the right represent question nodes. Arrows between nodes represent arcs defining possible transitions between nodes. For clarity, conditions on arcs have been omitted from this diagram.

In Q-MED, the nodes of the ATN are connected by *arcs* that represent *rules* for deciding which nodes to visit. For instance, we might label an arc "low back pain, not exacerbated by exercise"— such an arc will be traversed if low back pain is known to be present, and, if the pain is known not to be exacerbated by exercise. Arc conditions may include disjunctions as well as conjunctions. For instance, an arc condition could be "(low back pain or hip pain) and (improved by analgesics)." Arc conditions are typically labeled with lists of possible findings in the database, but also may be labeled with conditions used for special purposes (e.g., arcs may be labeled "nota," which stands for "none of the above.")

Q-MED stores all findings in a global finding tree [11]. Storing responses globally offers three main advantages over storing responses locally in the nodes. First, the ability to retrieve global data for testing arc conditions reduces the branching factor of the ATN and eliminates the need for redundant nodes.

Second, because findings deduced from different sub-networks are stored in a common database, answers to questions in one sub-network can affect the selection of questions in another sub-network. Allowing sub-networks to interact in this way is important to Q-MED, because Q-MED asks many open-ended questions whose responses may be relevant to many different sub-networks.

Third, if responses were stored locally with their associated nodes, the rules would have to include references to such nodes. For instance, let us suppose that a knowledge engineer would like to label an arc with "course intermittent and improved by analgesics." In the scenario using local storage, the knowledge engineer might write as a condition "response to node 100 is 'course intermittent' and response to node 124 is 'location hip.'" However, this method is problematic, because it may not always be possible to know where a particular finding is stored, especially in interviews in which not all questions are yes–no or multiple choice. The finding "improved by analgesics" might be stored in the any of three nodes: "Tell me about your pain," "What relieves the pain?" or "Does aspirin help to relieve the pain?" Therefore, it is crucial to Q-MED that the ATN formalism allow for the storage and retrieval of data in global registers.

In addition to questions and subroutine calls to sub-ATNs, nodes also contain *side effects*—commands issued when a node is visited. Side effects in Q-MED's ATN include conditions for bypassing nodes (*bypass* rules), instructions for driving the user interface, and instructions for storing data in the global finding tree. Conditions for bypass rules are specified in the same format as are arc conditions—a list of findings to look up in the database. If the conditions are met, the node is bypassed, and control is transferred to the next appropriate node. This functionality is used in Q-MED to avoid asking redundant questions. For instance, the question "Does aspirin relieve the pain?" would be bypassed if the finding "relieved by aspirin" were already stored in the global finding tree by a node visited earlier in the interview. The ability to attach bypass conditions to nodes relieves the knowledge engineer of the burden of specifying such conditions on all the arcs leading to the node.

In Q-MED, displaying the text of the question on the screen, switching speech recognition grammars, requesting input from the speech-recognition device, and parsing the input are all side effects associated with the nodes of the ATN for driving the interface. Once the input is parsed into particular findings, the findings are stored in the finding tree as a final side effect.

A TRACE THROUGH THE ATN

To better understand how Q-MED uses an ATN, it is instructive to walk through a simplified trace of the program through a portion of the back-pain ATN. Figure 2 shows a small piece of the back-pain ATN. The program begins at the concept node labeled "general." This node is a concept node and thus invokes an associated sub-ATN. The first question node in the sub-ATN that we encounter is, "Tell me about the pain you've been having." Let us suppose that the patient answers, "The pain is in my lower back." The parser extracts the finding "location low back" and stores the finding into the global finding tree. Next, we have a choice between traversing two arcs: one labeled "none of the above" ("nota"), and the other labeled "finished." "None of the above" and "finished" have special meanings. The "none of the above" arc is traversed if the conditions of no other arc emanating from the node are satisfied. The "finished" arc is traversed if the patient says "I'm finished." Because the patient said, "The pain is in my lower back," the program cannot traverse the "finished" arc; instead, it traverses the "none of the above" arc, leading to the node, "Please go on." Notice that this node is connected to itself via a "none of the above" arc. This loop allows the patient to enter as many findings as she likes, before she says, "I'm finished." Once the patient does say, "I'm finished," the "finished" arc is traversed and the program reaches a "return" node. This node signifies the end of the sub-ATN, and control is passed back up to the node labeled "general" in the higher level.

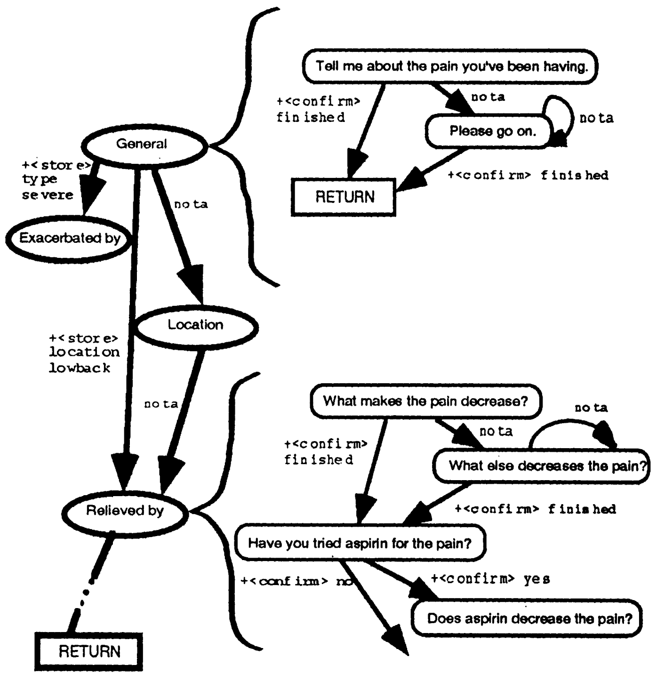

Figure 2. A small piece of the back-pain ATN. Concept nodes are represented by ovals, question nodes by rounded rectangles, and return nodes by plain rectangles. Arrows represent arcs, along with their associated conditions for traversal. Only the sub-ATNs of the concept nodes "General" and "Relieved by" are shown here.

At this point, three different arcs may be traversed, depending on the current status of the database. If the finding tree contains the finding, "type severe," then control is passed to the node labeled "exacerbated by." In this case, the database contains the finding "location low back," so the arc leading to the node "relieved by" is traversed. This node invokes the sub-ATN that asks questions about what activities or medications relieve the pain. The first question in this sub-ATN is "What makes the pain decrease?" As before, if the patient answers "I'm finished," control is passed to the node "Have you tried aspirin for the pain?"—otherwise, control is passed to the node, "What else relieves your pain?" Program execution continues in this way until we reach a "return" node at the highest level.

This example brings out two notable points about Q-MED's use of ATNs for representing interviewing knowledge. The first is that Q-MED can use a question multiple times in a loop. In Figure 2, the questions, "Please go on" and "What else decreases the pain?", are connected to themselves by "nota" arcs. This simple construct allows patients using Q-MED to give multiple responses to one question. If we desired, we could create more complex control constructs using multiple nodes and arcs.

The second point is that we can specify a main path through the ATN. Notice that in Figure 2, the concept nodes "general," "location," and "relieved by" are connected linearly by "none of the above" arcs. This linear construct forms a default path through the concept nodes of the ATN; unless the patient enters either of the findings, "type severe" or "location low back," Q-MED will follow this default path.

ATNS AS A GENERAL REPRESENTATION FOR HISTORY-TAKING SYSTEMS

The previous sections described how Q-MED uses an ATN to represent the interviewing knowledge for a speech-driven automated history-taking program that uses IF–THEN rules as a control structure, and a global finding tree for data storage. This section will discuss ways in which the ATN formalism could be used for interviewing systems that have different control methods, interfaces, and data-storage mechanisms.

Use of Different Control Methods

Q-MED uses IF–THEN rules to determine the sequence of nodes to be visited, where each arc between nodes represents a rule. However, the ATN formalism does not restrict tests on arcs to be IF–THEN rules. We could label arcs with scores representing probabilities or utilities to emulate a sequential Bayesian or probabilistic control method [7]. Alternatively, we could treat the cells in the statistical matrix of Warner's systems as global registers accessible by side-effect statements in the nodes and arcs.

To emulate a primitive branching control method, such as that used in the AMH [8], we could simply limit the conditions on each arc to the response of the question just asked, rather than allow the retrieval of global data for testing of arc conditions. We could even combine different control methods, perhaps employing different mechanisms for different levels of the ATN. For instance, we might use a probabilistic method to determine the order of sub-ATNs to call on a high level, but use simple IF–THEN constructs for ordering the specific questions within each sub-ATN on lower levels.

Use of Different Interfaces

Interface commands are issued as side effects associated with the nodes of the ATN. Therefore, we do not need to change the structure of the ATN to implement different interfaces for the interviewing system. For instance, to change the output modality from the text output of Q-MED to voice output, we simply modify the side effects of the nodes to play sound files instead of displaying text files. Likewise, changing input modalities does not affect the structure of the network. For instance, replacing the speech-driven input modality to a pen-based one would involve modifying the side effects of the nodes to call a pen driver instead of a speech driver, and modifying or removing the parser.

Use of Different Data-Storage Methods

Q-MED receives input from the speech recognizer, extracts relevant findings, and stores findings hierarchically in a global finding tree. Q-MED tests conditions on arcs by performing database searches in the tree. This storage and retrieval mechanism could be replaced with other mechanisms, depending on the requirements of the interviewing system. For instance, a system that limits questions to yes–no could replace the finding tree with a global array of questions and their corresponding responses: "yes," "no," or "unknown." Retrieval of data would involve simple indices into the array. A sequential Bayesian system such as Warner's could treat the statistical matrices as global storage areas; responses to questions at each node would result in updates to these matrices. Because the ATN formalism allows for arbitrary setting and reading of global registers, any data-storage and retrieval mechanisms could be used.

DISCUSSION

We identify three characteristics of ATNs that facilitate the use of ATNs in interviewing systems: explicitness, hierarchical structure, and generality.

Explicitness

ATNs as a representation for interviewing knowledge provide a simple, explicit framework on which knowledge acquisition can be structured. Graphically, ATNs resemble flowcharts for clinical algorithms found in numerous medical textbooks and journals. For a knowledge engineer, the translation of a flowchart to an ATN is simple. Decision nodes in a flowchart would be represented as nodes of an ATN, and branches from decision nodes would be represented as arcs (with its associated conditions).

Knowledge acquisition is facilitated also by the explicit nature of the nodes and arcs of an ATN. To clarify this idea, let us compare Q-MED's ATN representation with that of a hypothetical, analogous system that uses only production rules for its representation. In a system using only production rules, the rules define a rule tree, but this tree is implicit. The knowledge engineer does not explicitly define this tree, but rather adds rules incrementally to an existing set of rules that implicitly form a tree. The addition of rules causes changes—that are not immediately obvious—to the rule tree and to the resulting sequence of actions [12]. On the other hand, the ATN formalism forces the knowledge engineer to create explicit nodes and arcs in the network as he adds rules to the system. This explicit structure results in a simpler mapping between the rules and the desired sequence of actions of the interviewing program, facilitating knowledge acquisition.

Hierarchical Structure

Although ATNs resemble flowcharts, an important difference between the two representations is that ATNs are hierarchical. Miller, who used ATNs in expert critiquing systems such as ATTENDING, identifies two advantages that ATNs have over single-level flowcharts [13,16]. First, the ATN formalism forces the knowledge engineer to conceptualize knowledge in hierarchical terms. Miller found that the hierarchical model gave his critiquing systems an organized, useful structure for generating the prose critique [13]. For history-taking systems, the hierarchical model allows the knowledge engineer to group related questions into sub-ATNs that can be treated as one unit, independent of other sub-ATNs. For instance, a group of questions that together allow the system to deduce the severity of the patient's pain could be treated as a single question that could, in one instance, be bypassed by the control structure, or, in another case, be reused multiple times for multiple pains.

The second advantage of a hierarchical structure is that paths that diverge at a lower level rejoin the same path at a higher level. In Miller's ATTENDING system, in which arcs represent anesthetic techniques, and paths through the ATN correspond to anesthetic management processes, this advantage makes it natural to model linear decision making where each decision involves many choices [16]. Q-MED uses this advantage too. For example, in Figure 2, regardless of how far the two paths leading from the question, "Have you tried aspirin for the pain?" diverge from one another, they will eventually rejoin at the "relieved by" node on the higher level, once they reach a node labeled "RETURN." In a flat flowchart model, two diverging paths might never rejoin unless they are designed explicitly to do so. Thus, the ATN formalism allows the knowledge engineer to structure the interview along only a few paths, with temporary divergences off the main paths, rather than having to structure the interview along several complex, parallel paths.

A third advantage of using ATNs to represent interviewing knowledge is that the existence of multiple levels allows the interviewing system to use different control structures for different levels of the interview process. A high, abstract level might use probabilistic methods to govern the sequence of execution of sub-ATNs, whereas a lower, more detailed level could use simple IF–THEN control structures to determine the sequence of specific questions.

A final advantage is that the division of knowledge into multiple levels allows the knowledge engineer to make changes to one level without affecting another level. For instance, in Figure 2, we could change the ordering of the concept nodes "location" and "relieved by," without having to make changes to their associated sub-ATNs.

Generality

As discussed earlier, the ATN formalism's generality allows an interviewing system to use different control methods, interfaces, and data-storage methods. Changing such characteristics requires the developer to specify appropriate side effects.

CURRENT STATUS

We have developed two ATNs for Q-MED: one for interviewing patients who have abdominal pain, and the other for interviewing patients who have back pain. Both systems are speech-driven; both use identical control structures and data-storage mechanism. The operation and preliminary evaluations of these systems are described elsewhere [11].

SUMMARY

Though history-taking systems have existed for decades, no standard scheme for representing interviewing knowledge has emerged. The lack of a general, yet well-defined representation scheme may be, in part, due to the different, specialized requirements of such systems. However, from our experience in building Q-MED, we believe that the ATN formalism is sufficiently general to meet the needs of most interviewing systems in different domains and environments, yet well-defined and explicit to facilitate the construction of such systems.

We offer three insights for using ATNs as representation schemes for history-taking systems. First, the hierarchical structure of ATNs affords a useful organization to the interviewing knowledge and a clean framework for knowledge acquisition. Second, global storage allows for useful interactions between sub-ATNs of a network, and reduces the branching complexity of the ATN. Finally, side effects of nodes and arcs provide methods for adding power and generality to the ATN formalism.

ACKNOWLEDGMENTS

We thank Richard Lin and Smadar Shiffman for their help in developing Q-MED's parser. We also thank Lyn Dupre for help with the style of the paper, and Edward Shortliffe for providing an environment that supports this type of research.

Primary support for this research is provided under contract #213-89-0012 from the Agency for Health Care Policy and Research. Additional support is provided by the Department of Defense and the National Library of Medicine under grants LM-04864 and LM-07033. Computer facilities were provided by the SUMEX-AIM Resource (LM-05208) and through an equipment grant form Speech Systems, Inc.

References

1. Collen, M.F., et al., Automated Multiphasic Screening and Diagnosis. *American Journal of Public Health*, 1964. 54(5): p. 741-750.
2. Pynsent, P.B. and Fairbank, J.C.T., Computer Interview System for Patients with Back Pain. *Journal of Biomedical Engineering*, 1989. 11: p. 25-28.
3. Stead, W.W., et al., Computer-Assisted Interview of Patients with Functional Headache. *Archives of Internal Medicine*, 1972. 129: p. 950-955.
4. Slack, W.V., A History of Computerized Medical Interviews. *M.D. Computing*, 1984. 1(5): p. 53-59.
5. Friedman, R.B., et al., An Automated Verbal Medical History System. *Archives of Internal Medicine*, 1978. 138: p. 1359-1361.
6. Grossman, J.H., et al., Evaluation of Computer-Acquired Patient Histories. *Journal of the American Medical Association*, 1971. 215(8): p. 1286-1291.
7. Warner, H.R., et al., A Sequential Bayesian Approach to History Taking and Diagnosis. *Computers and Biomedical Research*, 1972. 5: p. 256-262.
8. Mayne, J.G., et al. Toward Automating the Medical History. Mayo Clinic. 1968. Rochester, Minn.
9. Johnson, K., et al., Q-MED: A Spoken-Language System to Conduct Medical Interviews. February 1992, Stanford University, KSL-92-09.
10. Bloom, S.M., et al., Converse: A Means to Write, Edit, Administer, and Summarize Computer-Based Dialogue. *Computers and Biomedical Research*, 1978. 11: p. 167-175.
11. Johnson, K., et al., Q-MED: A History-Taking System That Uses Continuous Speech Recognition. March 1992, Stanford University (in press).
12. Davis, R., et al., Production Rules as a Representation for a Knowledge-Based Consultation Program. *Artificial Intelligence*, 1977. 8: p. 15-45.
13. Miller, P.L., et al., HYDRA: A Knowledge Acquisition Tool for Expert Systems That Critique Medical Workup. *Medical Decision Making*, 1987. 7: p. 12-21.
14. Woods, W.A. Transition Network Grammars for Natural Language Analysis. *CACM*, 1970. 13: p. 591-606.
15. Barr, A. and Feigenbaum, E.A. *The Handbook of Artificial Intelligence*. Los Altos, CA: William Kaufmann, Inc, 1981.
16. Miller, P.L. *Expert Critiquing Systems: Practice-Based Medical Consultation by Computer*. New York City, NY: Springer-Verlag, 1986.