SYSTEMS ARCHITECTURE AND PATTERN RECOGNITION FOR AN INTELLIGENT, PEDIATRIC, NON-CONTACT COMMUNICATION AID

Daniel Cossever[1], Tom Chau[1,2], Deryk Beal[3], Brian Kavanagh[3], Eric Bouffet[3]

[1]IBBME, University of Toronto, [2]Bloorview MacMillan Children's Center, and [3]The Hospital for Sick Children

***Abstract* - The system architecture for an intelligent, pediatric, non-contact communication aid is proposed for temporarily mute patients. It consists of user profile, knowledge base and pattern prediction modules interfacing through a central information manager. The information manager communicates with the user via a dialogue manager and visual interface.**

To ensure that the system is easy to use, several learning algorithms are proposed for the pattern prediction module to be able to predict future inputs and display them for easy location and selection by the user. A backpropagation algorithm is explored in detail, where data triplets are used to train the network. The data is generated through a simulator that creates sets of data based on biases for time of day, previous selection(s), and time gap between selections. Although the prediction rate is heavily tied to the simulator biases (representing underlying behavioral patterns), it is significantly increased compared to random guessing. The prediction rate is also closely linked to the number of training cases that the network gets to see.

Implementation of such learning algorithms within the structure proposed may lead to improved speed and accuracy of communication, and less support required to teach and maintain the system.

***Keywords* - Augmentative and alternative communication, backpropagation, communication aid, data generator, pattern recognition, system architecture.**

INTRODUCTION

Temporary voiceless is a common problem encountered in intensive care patients. Causes range from surgical side effects to application of medical devices (e.g. intubations) (Costello, 2000), but effect is always the same: an inability to communicate basic needs. This is especially predominant in pediatrics, where pre- or emerging literate children cannot communicate through conventional writing or complicated gestures.

A communication aid is proposed that will allow pediatric intensive care patients to have access to digital visual stimuli that can be manipulated in order to obtain a desired output. In order to facilitate communication, it is important to make the interface between the system and the user as simple as possible with the basic requirement being only to have an appropriate output based on user input. Although the exact type of interface is unimportant here, as there are many methods of achieving such a requirement, it is assumed that the interface will translate patient movement into a selection of iconic representations that are translated into speech.

This paper starts with an outline of the proposed communication aid system architecture, followed by a more in-depth look at possible intelligent features, and the implications on system performance and patient interaction.

BASIC SYSTEM ARCHITECTURE

The system consists of several different modules which are controlled by a central information manager. The external system consists of user input and feedback and an external system administrator while the internal system consists of User Profile, Knowledge Base, Pattern Recognition, Timer, Dialogue Manager, and Visual Interface modules, all which feed into the central Information Manager (see Figure 1).

User Input and Feedback

To allow communication for patients that may have difficulty in using a conventional mouse (due, for example, to muscular atrophy, paralysis, or other such motor impairments) a web-camera will be used to translate patient movement (of any magnitude by any body part) into system inputs. The user will have visual feedback on how their movement is translated into system input to aid in navigation and selection of icons.

Visual Interface

The primary function of the visual interface is to convey user intention to the system through gesture recognition, which is the process by which the system attempts to robustly discriminate user motions and convert them into valid system input. The visual interface is also responsible for automatically determining the range of motion of the user (movement space) and adjusting the camera to compensate.

Dialogue Manager

The Dialogue manager is responsible for converting all inputs and outputs into their required form. The external inputs into the system (camera and system administrator) are converted into a digital form that allows for signal processing and storage. Conversely, system outputs (motion feedback, pattern prediction, etc.) are converted to a visual form as to be easily understood by the user.

Timer

Any camera input that is deemed to be an iconic selection by the visual interface will be time-tagged to keep temporal control and to help with classification and prediction of the inputs.

User Profile

A profile will be created for every new user of the system containing specific physical, historical, circumstantial, and preference information. All system interactions of that individual will be recorded here, and used in conjunction with profile information to aid in improving system interaction and prediction of future inputs.

Knowledge Base

The knowledge base is a collection of information that applies more or less universally. Initially, the information contained in the knowledge base will only consist of human physiological requirements (e.g. will eat and sleep every day), but will be increased in breadth the more the system is used. Once a minimum threshold has been reached within user profiles that show similar information on a specific subject, the knowledge base will be updated to include that information. This is to reduce the system learning and adaptation time to each new user.

Pattern Recognition

The Pattern Recognition Module consists of the prediction of inputs (discussed in detail in the following section), as well as memory reduction techniques (to limit the amount of information storage space required) and decisions on updating the visual interface to ease interaction.

Information Manager

This processing unit controls all aspects of the communication aid, which can be broken down into two main categories, module communication and system maintenance.

Module communication consists of relaying all internal system information to the required modules. It is also responsible for assimilating information from all modules and outputting to the Visual Interface accordingly.

System maintenance ensures that the internal environment functions optimally. Responsibilities include managing storage space, ensuring system stability when making alterations to the system, and information matching to ensure reliability and consistency of all information being passed.

External System Administrator

This module provides access to the system from an external administrator to add or update User Profiles and Knowledge Base, as well as change system settings for the Information and Dialogue Manager.

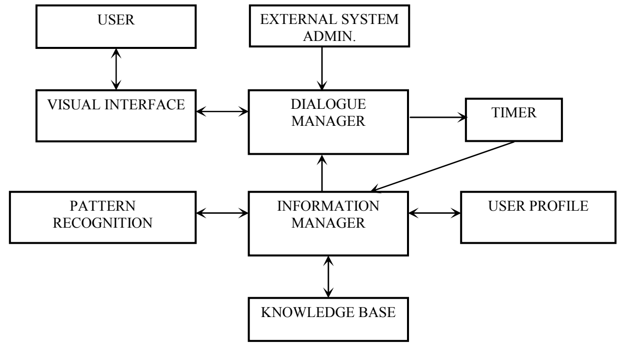

Figure 1 – Basic Communication Aid System Architecture

Brain Analogy

The manner in which the system interprets inputs is fairly intuitive since it is similar to the way that humans assimilate information from their environment. The brain processes massive amounts of rapidly changing perceptual information and tries to categorize it according to its current perception (pattern recognition). The basis of that perception is largely related to our personality and previous encounters with similar information. If enough encounters occur that are contrary to our perception, then that perception might change (user profile). If enough information is processed that cannot be adequately represented by our current perception, the underlying basis for that belief may eventually be altered (knowledge base).

PATTERN RECOGNITION

One of the major criteria for an intelligent pediatric communication aid is that is simple to learn and simple to use. This module contains algorithms by which previous user inputs are used to predict currently desired ones, to reduce icon location and selection times. The simplest type of prediction is a frequency counter, whereby the most often selected icons are the most likely to be selected in the future. Other more in depth algorithms (learning networks) include a backpropagation algorithm to predict inputs based on the immediately preceding selections, and a Markov model to predict inputs based on the historical sequences observed.

A backpropagation algorithm was implemented in order to ensure that learning networks as such can improve the interaction with the system. In this case, a fully connected network was used with one layer of hidden units and small, random initial weights and biases for the hidden and output units (see Figure 2). Within the backpropagation algorithm, momentum is used to speed learning, and a weight cost is used to keep the weights small. The network gives an input dependent output for each icon, (where the largest output corresponds to the best prediction) and a cross-entropy error (the negative log probability that the network gives the right answer for each case).

The major implementation difficulty is that a substantial amount of data is required in order to train the network to the point of having noticeable improvement. As no patient data of this type has yet been collected, a simulator was created to generate data.

Data Simulator

A user model has been created in an attempt to replicate the inputs the network might see with a 'typical patient'. Each set of icons consists of the icon to be predicted (icon 3), and the two previously selected icons (icon 1 and icon 2).

The time of day that each set occurs is random, and the time gap between icons in a set is based on probability tables (smaller gaps being more likely than larger ones). Bias table were created (with efforts to make them as realistic as possible) to alter the probabilities that each icon was selected based on time of day and previously selected icons. Each possible icon was put through a normalized softmax (see Equations (1), (2), (3)), and selected randomly based on their percent probability from that softmax. The softmax used to generate icon 1 probabilities is shown in Equation (1).

$$P_i = \frac{\exp(\mathrm{b}_{td})}{\sum_{icons} \exp(\mathrm{b}_{td})} \tag{1}$$

P_i: probability of selecting icon i
b_{td}: bias from time of day
b_{1p}: bias from previous icon selected

The softmax used to generate icon 2 probabilities is shown in Equation (2).

$$P_i = \frac{\exp(\mathrm{b}_{td} + \mathrm{b}_{1p} + \mathrm{f}_{tg1p})}{\sum_{icons} \exp(\mathrm{b}_{td} + \mathrm{b}_{1p} + \mathrm{f}_{tg1p})} \tag{2}$$

f_{tg1p}: function of the time gap since the previous icon

The softmax used to generate icon 3 probabilities is shown in Equation (3).

$$P_i = \frac{\exp(\mathrm{b}_{td} + \mathrm{b}_{1p} + \mathrm{f}_{tg1p} + \mathrm{b}_{2p} + \mathrm{f}_{tg2p} + \mathrm{b}_{1\&2p})}{\sum_{icons} \exp(\mathrm{b}_{td} + \mathrm{b}_{1p} + \mathrm{f}_{tg1p} + \mathrm{b}_{2p} + \mathrm{f}_{tg2p} + \mathrm{b}_{1\&2p})} \tag{3}$$

b_{2p}: bias from 2nd to last icon selected
f_{tg2p}: function of the time gap since 2nd to last icon
$b_{1\&2p}$: bias from combination of last 2 icons

The function used for time gaps is shown in Equation (4).

$$\mathrm{f}_{tg} = \begin{cases} \exp(-\log(\text{time\_gap}) * 0.2), & \text{for time\_gap} > 11 \text{ min} \\ 0.62, & \text{for time\_gap} \le 11 \text{ min} \end{cases} \tag{4}$$

This function is used so that the influence an icon exerts on the selection of the following icon increases as the time between them decreases. The cutoff at 11 minutes ensures that immediately preceding icons do not over-exert influence on the following selection.

In order to reduce the scope of the simulator, there have been several assumptions made, namely,

- System outputs represent those desired by the user (no incorrect or accidental outputs)
- There are only 10 possible outputs (plus a 'no output' selection)
- Only selections made within 3 hours of the output to be predicted have an effect on that output
- Only the 2 previous selections have an effect on the output to be predicted, regardless of time
- If there have not been 2 previous inputs in the 3 hours prior to the output to be predicted, the 'missing' outputs are considered to be 'no input'
- Requests by the user through the system are fulfilled

Learning Algorithm

Icon 1, time gap between icon 1 and 3, icon 2, time gap between icon 2 and 3, and time of day of icon 3 were used as inputs into the network for each set of data. During training, the output of the system was compared to the actual icon 3, and the error derivatives were backpropagated through the system to update the weights and biases (see Figure 2). After each iteration, separate testing data was run through the network (with no indication of the correct icon 3 output) to determine how well the output could be predicted (from choosing the icon with the maximum output from the network as the right answer).

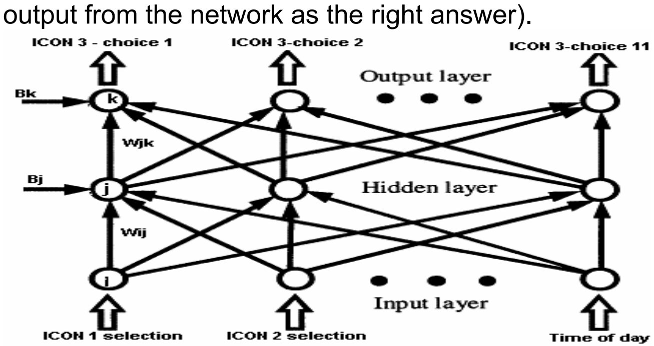

Figure 2 – Backpropagation Network (some components removed for clarity), where Wij is the connection weight between unit i and j, Bj is the bias of unit j.

Factors Affecting Prediction

The largest determining factor in how well the system could predict desired outputs was the magnitude of the biases (representing the strength of the underlying behavioral patterns). A bias dividing factor was used to illustrate this dependency, and the correlation between prediction rates and the bias dividing factor is shown in Figure 3. It must be noted that biases would change with every individual, and it is impossible to gauge where an 'adequate' bias is achieved. Imposing a constraint that the most probable selection is no more than 30 times more likely than the least probable selection (for icon 1) gave a bias diving factor of around 18, which was used for all simulations.

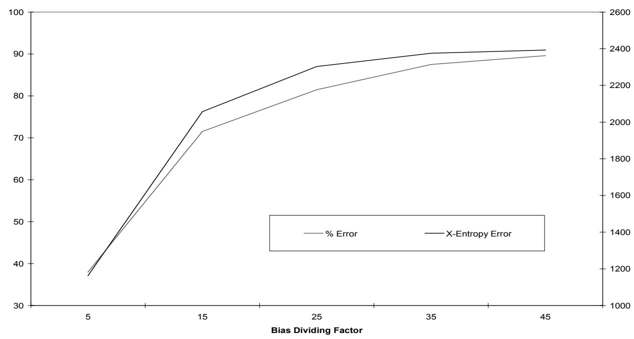

Figure 3 – Prediction Error Dependency on Bias Dividing Factor

Another significant determining factor in the accuracy of predictions was the number of training cases supplied to the system. Since each patient is different, and may only be using the communication aid for a limited period of time, it would be ideal to pre-train the system using simulated data. Unfortunately, it is extremely difficult to replicate all the different behavioral factors involved in accurately creating such a simulator. Training the system from an initial untrained state gave prediction rates shown in Figure 4. The network only improved the percent error rate from 91% to 90% after 250 training sets, and settled to a minimum of about 71% error rate after around 4500 training sets.

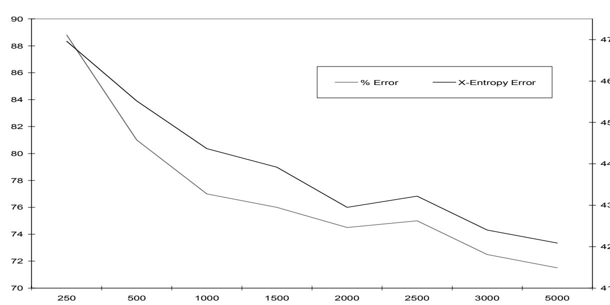

Figure 4 – Prediction Error Dependency on Number of Training Cases

Adjusting the algorithm parameters affected the ability to predict outputs, but given that the parameters were within a fairly broad range, did not have nearly as much of an effect as training cases or bias factor. The best results occurred with 20 hidden units, a slow learning rate and relatively large prejudice for keeping the weights small.

<u>Time Implementation</u>

Time of day is represented by a series of 12 triangles, each with peaks 2 hours apart. The base of each triangle spans 6 hours, meaning that only if the time of day was ±3 hours from each peak time would there be any input for that triangle. The input for each triangle is the ratio of height of the triangle at the specified time to the height of the full triangle. This gives a more consistent flow to the time of day, where nearby times have similar inputs to the system. Although specific time would have been adequate for the backpropagation algorithm, having nearby times give similar inputs will help when implementing a learning algorithm that predicts inputs via sequences from similar times of previous days.

CONCLUSIONS AND FUTURE WORK

The implementation of a backpropagation algorithm can definitely improve the prediction ability of an intelligent, non-contact communication aid, but the degree depends on the intensity of the underlying personality patterns, and the amount of exposure an individual has with the system.

Future work includes adding information from the Knowledge Base and User Profile as inputs to the learning networks, implementing the Markov model and frequency counter, and optimizing the Information Manager so that when different predictors output the same prediction, then one of them would revert to the second best option. The intelligent communication aid will then be tested against a non-intelligent counterpart to quantitatively assess the actual benefit of intelligent features.

It is expected that the application of learning algorithms and other intelligent features to a communication device will decrease frustration levels and familiarization time with the system, and increase the speed and accuracy of patient communication. It is also expected to require less maintenance and setup by health care professionals, as the system can be automatically tailored to each individual user.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Geoff Hinton for providing material for the backpropagation algorithm, the Bloorview Macmillan Children's Center and PRISM lab, the University of Toronto, and The Hospital for Sick Children.

REFERENCES

Costello JM (2000), AAC interventions in the intensive care unit: the Children's Hospital Boston model, <u>AAC Augmentative and Alternative Communication,</u> 16:137-153