

Award Number:
W81XH-09-2-0158

TITLE:
Developing Multi-Voice Speech Recognition Confidence Measures
and Applying them to AHLTA-Mobile

PRINCIPAL INVESTIGATOR:
Dr. Greg Gadbois

CONTRACTING ORGANIZATION:
HandHeld Speech
Amesbury MA 01913

REPORT DATE:
May 2011

TYPE OF REPORT:
Final Report

PREPARED FOR:
U.S. Army Medical Research and Materiel Command
Fort Detrick, Maryland 21702-5012

DISTRIBUTION STATEMENT:
Approved for public release; distribution unlimited

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision unless so designated by other documentation.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 15-05-2011		2. REPORT TYPE Final		3. DATES COVERED (From - To) 15-07-2010 - 15-04-2011	
4. TITLE AND SUBTITLE Developing Multi-Voice Speech Recognition Confidence Measures and Applying them to AHLTA-Mobile				5a. CONTRACT NUMBER W81XH-09-2-0158	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Dr. Gregory J. Gadbois				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) HandHeld Speech LLC 18 Hillside Ave Amesbury MA 01913-2206				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Med. Research ACQ Activity U.S. Army Medical Research and Materiel Command Fort Detrick, Maryland 21702-5012				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT While developing a documentation solution for medics in the field, confidence measures have been incorporated to improve command rejection and minimize enrollment. The new enrollment algorithms makes speaker-dependent recognition competitive with speaker-independent recognition.					
15. SUBJECT TERMS Hands-free eyes-free speech recorder; Confidence measures in speaker-dependent speech recognition enrollment, and in command rejection;					
16. SECURITY CLASSIFICATION OF: U (unclassified)			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 32	19a. NAME OF RESPONSIBLE PERSON USAMRMC
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (include area code)

Table of Contents

	<u>Page</u>
Introduction.....	4
Body.....	5
Confidence Measures.....	5
Hands-Free Recorder.....	8
The Medic's ability to Multi-Task.....	8
The Solution to Field Documentation has Four Parts.....	9
The Recorder's User Interface...,.....	10
The Wind.....	13
Pros and Cons of the Throat Mic.....	13
Confidence Measures in Enrollment.....	14
A Catalog of Models.....	16
GumStix.....	17
A Viewer.....	18
The Final Test, OD3.....	18
Key Research Accomplishments.....	19
Reportable Outcomes.....	19
Conclusion.....	20
References.....	20
Appendix A: Details of the Fit / Confidence Measure Calculation.....	21
Appendix B: Hands Free Voice Recorder User Manual.....	23

Introduction:

A confidence measure would be very useful in speech recognition. With a dependable confidence measure we could know when something has gone wrong in the recognition process and we could know when not to trust the recognition answers. We could correctly reject recognition results. Good confidence measures are difficult to create in speech recognition. The problem is that recognizers barely get right answers. "Right" results do not stand out head and shoulders above wrong results. In the open dictation problem, right results are indistinguishable from good scoring garbage.

At HandHeld Speech, we are less interested in open dictation, we are interested in constrained speech problems (command and control, database access, etc.). The interesting question is how would some of the ideas that have failed as open dictation confidence measures fare when applied to a constrained recognition problem. In our original proposal and statement of work we proposed to look for good confidence measures in these problems. By the time this proposal was funded, more than a year had gone by and we had preliminary results. In the first months of this contract we finished the confidence measure research. Then we negotiated a new statement of work reflecting the fact there was much less research to do. Independently, our sponsors were involved in a JMDSE program with a need for a speech recognition application. Our sponsors had been tasked to see if speech recognition could be use by medics in the field to better document casualty care. Our new statement of work included a major new piece, the participation in the JMDSE task.

As an aside, our participation in the JMDSE program was very different from most of the other participants. The JMDSE program was set up as an inquiry, to look at existing devices in the civilian healthcare market, to see if they could become useful to medics in the field. Most of the other participants were making small modifications to shipping products. We had no such shipping product. In fact there were aspects of the task that were a research problem. The culmination of the JMDSE program was a technical demonstration where high level officers were invited to see devices used in lifelike tests. The purpose was the education of those officials of existing products in the civilian market and how they might fit into military use.

The last thing in this introduction is the new statement of work. The major points are:

- 1) We would finish the confidence measure research.
- 2) We would use the confidence measures to improve rejection of extraneous speech.
- 3) We would apply the confidence measures to the enrollment process in the hope of making a very short enrollment.
- 4) We would build a hands-free recorder for form filling and an accompanying viewer of that recorded data (the JMDSE task).
- 5) We would would port a version of that recorder to embedded Linux, to run on gumstix's (embedded hardware, see www.gumstix.com).

Body:

Confidence Measures:

At the start of this contract, a major part of the research on confidence measures was finished. We were testing and making minor improvements to the measures. For clarity, I will review the confidence measures we had earlier created and describe the particular improvements done in this contract. I will not describe the initial exploratory work.

Conceptually, a recognizer starts with a big list of all the things a user might say. When a person speaks, all the recognizer does is rank the list. It searches for the best fit. It compares the sounds that were heard to the way something on the list would be expected to sound. It does not hypothesize something outside of the list.

For command and control problems, we create a list of all the commands. In addition, we create word sequences that are not commands and put them on the list. We call these additional word sequences "the garbage rule". When the recognizer selects one of these word sequences as the best fit to an utterance, we know the user did not say a valid command. We can reject the utterance. This is how we cope with a user where in the midst of running an application he turns to a friend and says "how about those Red Sox" (and the recognizer overhears the conversation). We make the valid commands compete with garbage strings. When a garbage string wins (is the top choice in the choice list), we reject the utterance. A garbage rule, is a type of confidence measure. When a garbage string wins we know the valid commands have low confidence.

Generally recognizers allow regular expression like syntax to specify list items. In the syntax of the HandHeld Speech recognizer, here is a rule that makes "legal" all possible sequences of spoken digits:

RULE Digits ({ 0 1 2 3 4 5 6 7 8 9 } +);

The '+' indicate one or more of the items selected from the previous set.

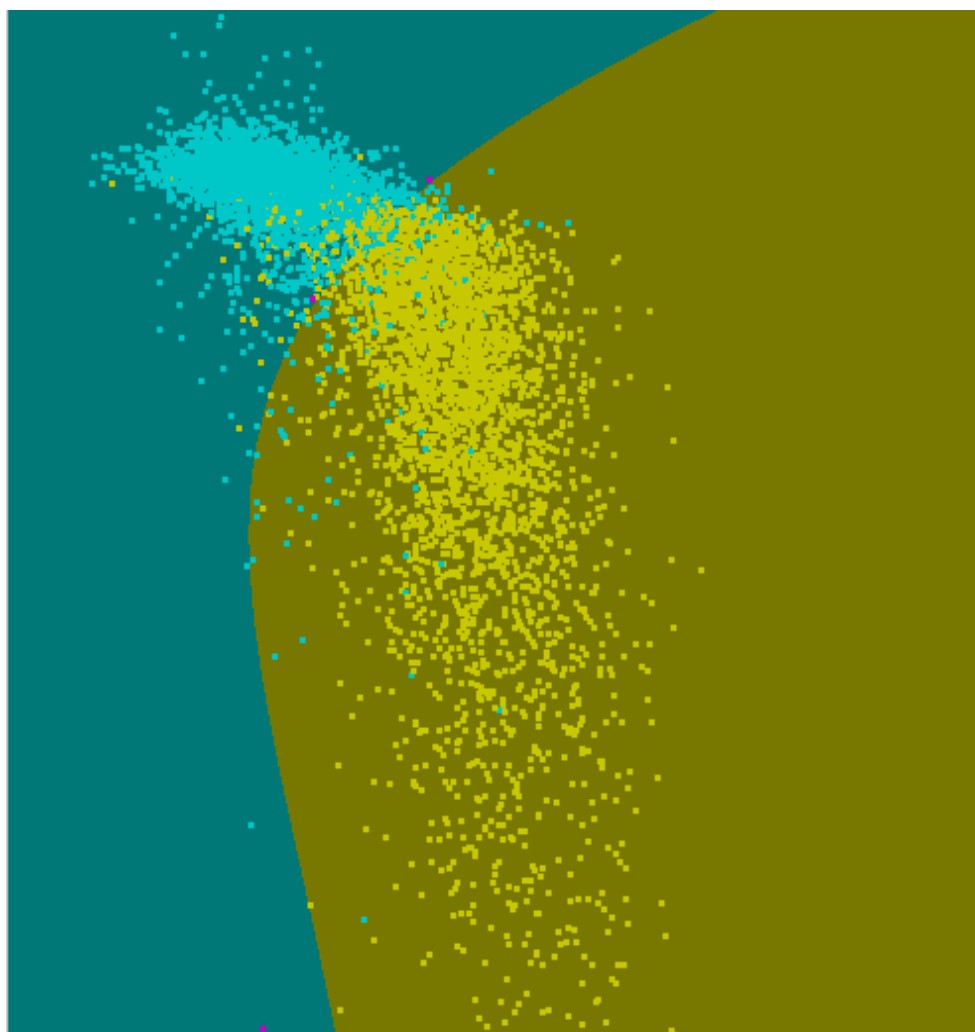
Similarly we can create a garbage rule, replacing the { 0 1 2 3 4 5 6 7 8 9 } with a set of one syllable words that span the set of vowel sounds in English. Then when a user says some random thing, there is a sequence of garbage words approximately rhyming with the utterance and winning in the recognition problem. The near rhyme is a much better fit than one of the few valid commands. The garbage rule is the first line of defense for rejection purposes, if the garbage rule is the best fit to an utterance, we know to reject it. It is not a valid command.

When a valid command is the best fit for the utterance (it beats the garbage rule) there is a further confidence measure to be made.

We calculate two quantities, (x, y), based on how the sound data fit the hypothesis on a frame-by-frame basis. (A frame of data is approximately 200 millisecs of digitized sound.) How the detailed fit varies is data on the accuracy of that choice. x is an average score per frame measurement and y is a duration penalized average score per frame. We give a complete description of these quantities and how they are calculated in Appendix A.

To get a feel for how these measurements might be used we made scatter plots. We plotted an (x,y) point in one color when it pertained to a right recognition, and in another color when it was the details of an incorrect hypothesis. With our measures, we are able to see the data separate into clusters where the “good” (measurements made on correct valid recognitions) are displaced from the “bad” (measurements where the correct transcription was not allowed into the list of possible hypotheses).

Below is a graph of real data points. The “good” and the “bad” separate fairly well into clusters with (small overlap). In the graph the “good” points (valid commands recognized) are cyan and the “bad” points (typically non sequitur utterances recognized as a rhyming string of one syllable words) are yellow. (To make the graph below we used LIBSVM, a Support Vector Machine software package¹)



Scatter plot of “good” and “bad” recognitions

¹LIBSVM is provided by Chih-Chung Chang, and Chih-Jen Lin, Department of Computer Science and Information Engineering, National Taiwan University.

Looking at many plots from multiple users under varying degrees of noise, we find the absolute positions of these clusters are dependent on many things. Both the state of the models (how well the phonetic models were enrolled) and the level of the current background noise will shift the locations of the clusters. We saw no a priori method of locating these cluster centers. The obvious solution is to use a sampling of the clusters to track their locations. We do that. We use a history of “good” and “bad” utterances to track the clusters. In the absence a history, we rely solely on a garbage rule. But once we have seen both valid commands and rejected utterances, (relying solely on the garbage rule) we make an additional demand for the top choice to be accepted as valid, its primitive measures must be nearer to the location of good data points than the location of bad ones. Otherwise, we reject it.

So the algorithm is:

If the garbage rule wins, we know the utterance is to be rejected, and the details of the fit can be used to track the location of the “bad” cluster.

If a valid command wins (beats the garbage rule), in the absence of history, we accept it as a good utterance and add it to the history of “good” utterances (use it to track the “good” cluster location). If we have a history, it must also pass the test that its scores like a “good” utterance, and is closer to the “good” utterance cluster than to the “bad”. If it passes this additional criteria, we declare it “good” and use it to evolve the location of the “good” cluster. If it scores like a “bad” utterance, it is rejected and can be used to evolve the location of the “bad” utterances cluster.

Now we can explain the improvement we made to the rejection algorithm in this contract. The improvement is in the decision process of what points should be used in tracking the cluster centers. Generally the criteria was, when the top choice of recognition was deemed valid (it was not the garbage rule and its primitive measures were closer to the good cluster center than to the bad) the top choice was used to track the good. Otherwise it was rejected and that choice was used to track the bad.

The flaw that was noticed was that when an utterance is rejected (the garbage rule won) that does not necessarily mean that the fit of the top recognition choice was bad. For example, among the one syllable words in the garbage rule were the digits. If a digit string was spoken the utterance is properly rejected (a simple digit string was not a valid command). Yet the fit of the top choice may have been stellar. Using the garbage rule top choice to influence where the bad cluster center is located is not always a good idea. The change we made was when a recognition is to be rejected, we re-measure the fit using the worst choice on the choice list. We use that measure to track the bad cluster location. The worst choice on the choice list will always be a mediocre fit.

The final result, when we use this confidence measure as additional criteria for rejection, we gain a factor of 90% to 95% improvement in separating valid commands from invalid ones (depending on the problem). We get approximately a factor of 10 to 20 improvement in rejection. So, for example, suppose running a garbage allows you to reject 95% of the invalid commands, 5% of the invalid commands are incorrectly accepted as valid. With the addition of the “fit based” confidence measure, the rejection errors drop to between .5% and .25%.

When a garbage rule is augmented with a fit confidence measure, valid commands are well separated from invalid commands. User frustration is greatly reduced.

Hands-Free Recorder:

Chronologically, the next thing we worked on was the hands-free recorder. We participated in a series of tests, simulating the field use of the devices. We were given the opportunity to develop a strategy of how the application should operate, write that application, then see informed users simulate real use. Our attempts were mostly perceived as a string of failures, which they were. But it was also a very rapid learning experience. We will touch briefly on the failures, and then, more interestingly, what we learned about the task from them.

Some of the failures had to do with the particular hardware we were using. Connectors (the mic jack, the radio sleeve) were a major issue. They were a chronic point of failure. The final field hardware must have good connectors. We will not dwell further on this point.

Some of the failures had to do with the immaturity of the Starix radio. Their hardware and device drivers were evolving throughout the program. This was an unavoidable source of delay, their technology is new.

In general it was agreed that the hardware was purely a prototyping tool. A deployable solution must be much smaller and lighter. The simulators were told this and were instructed to concentrate criticism on the use and feel of the software.

The remaining failures caused user frustration, and observing that frustration was critical to the development of the application. The frustrations were:

- 1) complexity of the user interfaces
- 2) wind
- 3) enrollment

There were three significant developments arising out of these frustrations. We list them, then explore them in detail:

- 1) an appreciation of the medics ability to multi-task
- 2) throat mics (coping with the wind)
- 3) using the confidence measure during enrollment

The Medic's ability to Multi-Task:

The first implementations of the recorder were conceived as something close to form filling.

The TC³ card is the current documentation required of the medic. There is a section dealing with medications, another for vital signs, etc. The recorder was presumed to be filling out this card. There was an idea of moving about the form where you first said a keyword like “vital signs” to take you to that area, then said “blood pressure 140 over 70” to fill in the relevant field. The application was very much modeled on that imaginary form. As the application was exposed to simulation, it became clear, complexity was an issue. And the application was made simpler and simpler. What was unappreciated at the outset was the level of concentration the medic is applying to the first aid task. He has no time and only irritation for anything that distracts him from his primary mission. He has no ability to multi-task. His concentration is total. We realized, the documentation process can not require any thought. The input must be free form. The final version of the application had basically two commands, a turn on the recorder and a turn off the recorder command. We firmly believe this is the right user interface. Any other will cause frustration and the medics will not use it. Documentation is not their primary mission, patient care is. An activity that distracts from patient care will not be embraced. More complicated user interfaces will be shunned.

The realization of what the user interface must be for the medic’s task is one of the most important research results of this contract.

We next give our perspective on what we believe is the right shape of the total documentation solution.

The Solution to Field Documentation has Four Parts, they are:

There is a simple field recorder capturing the data. There is the transport of the digitized sound out of the field. There is the immediate use of the digitized sound. There is a final transcription process to permanent records. We list these points then give a discussion of each:

- 1) Capture: The fielded device needs to be small, light, and have long battery life. The only requirement is that it records digitized sound. It does not do form-filling. It records free form dictated notes.
- 2) Transport: Many copies of the digitized sound should be made and it should be sent out of the field, through many routes. The medic should retain a copy; a copy should stay with the patient; a copy should move forward into the system ahead of the patient.
- 3) Oral use: Oral records should become a normal part of patient records. Where a TC³ card is currently being referenced, oral records should be equally accessible.
- 4) Transcription: There is a final transcription for file purposes. The original recording should be retained.

Beginning with the first point: the limited ability of the medic to concentrate off task is why there can be no user interface more complicated than a simple recorder (no form-filling). Speech recognition enables a hands-free / eyes-free use of that recorder. Given such a limited recognition problem, it should be possible to create a tiny single purpose recognizer that has low computational needs; that would have both simpler hardware and longer battery life than what was fielded in our prototype.

The second point: A wireless method is preferred for bridging hardware in the movement of files. In the field, that wireless technology must be both low power (long battery life) and undetectable. There are two such technologies that I am aware of. One is the spread spectrum radio provided by Starix. The other is a technology called Near-Field Magnetic Induction, currently developed by Freelinc/Radeum. We tested the solution provided by Starix.

Starix provided a radio attachment for an MC70 pda and a USB memory stick device which can be accessed wirelessly through their radios or by plugging into a USB port. We were using these devices in our concept application. The recorder would copy wirelessly to the EIC (Electronic Information Carrier – the memory stick). The EIC was transported with the patient. The recorder would retain a copy of the digitized sound files (for the medic), and the EIC would evacuate with the patient. While in the transport vehicle, the EIC could be scanned. The files could be sent ahead to the field hospital using the vehicle's radio.

The third point: at the field hospital, someone could listen to the recordings in preparation for the arrival of the casualty. This would be a new procedure. The first reaction to this suggestion was that no one would listen to recordings. But this is a reasonable suggestion. When only the important statements about the care are recorded and all the silence between sentences is removed, the length of recordings is remarkably short. It would take less than a minute to hear all the statements. It must be made very easy to access these recordings. The benefit of listening to the recordings is that the attendant is informed directly by the caregiver. There are no errors introduced by either hearsay or a transcription process.

The final point: The field is not an office environment. The recordings coming from the field are a challenging speech recognition task. At today's level of technology speech recognition does not provide sufficiently accurate transcription for medical records. Currently people must be involved doing transcription. But that data (the recordings and their transcriptions) are what are needed for further research. With it, a serious research effort can follow to replace human transcription with automated speech recognition based transcription. I consider human transcription the first step in creating the fully automated solution. I believe a fully automated solution is a reachable goal.

Next we look at the delivered prototype, its user interface.

The Recorder's User Interface:

The Recorder is meant to be operated eyes and hands free. The delivered system is software running on an MC70 with a throat mic and Starix radio attachment. It writes files to an EIC (Electronic Information Carrier – Starix Memory Stick device). Accompanying this report is a User's Manual for that software (see Appendix B). Here I synopsize the important speech driven parts of the User Interface.

After enrollment, running the Recorder, there are 4 commands. They are:

- 1) WakeUp WakeUp WakeUp
- 2) WakeUp WakeUp New Patient
- 3) Go To Sleep
- 4) Train Last Command

There is an on screen mic button that will turn the mic on and off. Generally one never uses it. The mic is always on. It listens to everything you say and ignores it all, until you tell it to wake up.

There are two wake up commands. When first presented with a new patient or switching between patients, you say "WakeUp WakeUp New Patient". This command does two things, it turns recording on and it initiates a connection with the closest EIC (Starix Memory Stick device). When recording data about a patient, the Recorder needs to know where to store the recordings. When you say "WakeUp WakeUp New Patient", the MC70 scans (using the Starix radio) for the closest EIC and connects to it. The user gets confirmation of which EIC is found, he will hear the EIC id (the 4 digits printed on the EIC) read back to him. Subsequent utterances are then collected and stored both on the MC70 and the EIC. (The medic retains a local copy and a copy accompanies the injured soldier.) The silence between utterances is discarded. The medic is free to say anything he wants. When he says "Go To Sleep" recording is stopped and the recorder returns to the dormant state. Later he can either say "WakeUp WakeUp WakeUp" to use the previous EIC. Or he can say "WakeUp WakeUp New Patient" to scan for the closest EIC. While dormant (sleeping) the MC70 is actively listening to everything spoken and discarding the utterances (until it determines a wake up command is heard).

As an example scenario, suppose a medic has one injured soldier. He begins documentation by approaching the patient (so that the closest EIC is the one on the patient). He glances at the EIC and says:

"WakeUp WakeUp New Patient"

He hears the digits spoken back (they match the digits on the EIC). He then says:

"Gun shot wound to the right arm. Pressure dressing applied."

He will hear a little ding. It is an acknowledgment that the utterance was captured, everything is operating correctly. He does not have to note the time, the application records the time and date automatically for each utterance. He is temporarily done. He says:

"Go to Sleep"

He hears a two toned "ding dong" confirming the recording is stopped. He speaks to the patient and measures blood pressure. The application ignores the utterances, there are no beeps. He says:

"WakeUp WakeUp WakeUp"

He hears the same digit string as before as confirmation that recording is restarted and the recordings will be recorded to the same EIC as before. (There was no new search for the closest EIC – 'WakeUp WakeUp WakeUp' implies use the last known EIC.) He says:

"Blood Pressure 140 over 70"

He hears the confirming ding, then says:

“Pulse 65”

hears another confirming ding, and finally says:

“Go to sleep”

He hears the “ding dong” confirming recording is stopped.

That is the typical usage. The user uses one of the 2 WakeUp commands to turn the recorder on. Then he says whatever he wants to say however he chooses to say it, and utterances are recorded. Finally, he turns recording off by saying “Go to sleep”. Turn it on, say what you want, turn it off. Usage is very simple and does not distract the operator from the primary job of patient care. Documentation can happen even during hectic care giving.

There is one other oral command. Suppose the user said “WakeUp WakeUp New Patient” and nothing happens, there is no confirmation of the command. The application rejected the utterance, it did not believe it was a valid command. Suppose the user repeats the command, saying “WakeUp WakeUp New Patient” and this time the digit string of the closest EIC is repeated back. The user immediately says “Train Last Command”. He will hear “trained” spoken back to him. What has happened is the last utterance which the application recognized as “WakeUp WakeUp New Patient” has been used to adapt the users voice models. The phonetic models that were created during enrollment had a description of what “WakeUp WakeUp New Patient” should sound like. The actual “WakeUp WakeUp New Patient” that was just heard is compared to the description and the description is adjusted to look more like the last utterance. The phonetic models are moved in the direction of the last utterance. This adjustment is done cautiously. The difference between the model and the actual utterance is measured and the models are move 10% of the way toward the utterance. It may take a few instances of training to make a serious change in the models. But correspondingly, if the operator says something a little wrong and forces training, the models won't immediately become unusable.

The “Train Last Command” command allows the user to selectively train his models. If the application is not working perfectly (the operator feels like he is fighting with it to recognize a particular command) he is empowered to “fix it”. He can train his models on the fly with real utterances under real conditions. He can make the system work. This empowerment feature is an important part of user satisfaction.

To summarize, the basic operation is to turn the recorder on, say what ever you want however you want to say it, and then turn the recorder off. All interactions are verbal with both eyes and hands free. And there is audible feedback so that it is clear recordings are properly occurring. This is the user interface that the medic experiences. It is non-intrusive and so simple that the medic is not distracted from the primary mission of patient care.

From seeing how previous versions of the systems failed, I believe the User Interface must be this simple. Otherwise, the medics will reject it because it is a distraction and an irritation to

performing patient care.

The Wind:

During one test demonstration, the wind blew. In high winds conventional microphones become useless for speech recognition. The turbulent flow of the wind over the microphone, although not very loud, is directly on top of the active mic element. The magnitude of the wind signal can easily be larger than voiced speech. In high winds, the wind continuously triggers utterance detection. The spectrum of the noise is broad. In high winds, the recorder was utterly useless.

That the noise spectrum has a typical shape (as a function of wind speed) gives hope to the idea that signal processing will be able to remove a large part of the wind signature. But that would be a research project in its own right. We needed a more expedient solution for this contract. Instead we looked to unconventional microphones. We purchased throat mics from a variety of vendor and also one bone conducting ear mic. After preliminary testing, we chose the IASUS throat mic as the most promising candidate for a near term solution.

Pros and Cons of the Throat Mic:

Throat mics work by pressing a sensor directly against the neck. If you touch your hand to your neck and speak, you can feel the vibrations. The sensor picks up these vibrations just like a normal mic detects the air pressure variations of sound.

The good: the sensor is detecting the vibrations passing through the tissue of your neck, a semi-liquid substance about the density of water. Air has a much lower density. A throat mic has very bad impedance matching properties with respect to air. Environmental sounds traveling through the air have little affect on the sensor. Irrespective of the current environmental sounds, the mic appears to be operating in quiet conditions. Conclusion: there is no degradation of recognition accuracy due to environmental noise (there is no apparent environmental noise).

The bad: the sensor is detecting the vibrations passing through the tissue of your neck, the signal is muffled and attenuated. Particularly, the affect of lips is remote. The signature on sound due to lip position (lip rounding) and sounds that are generated near the lips are much less visible in the signal. There is lower information content in the signal. Recognition accuracy is degraded. Conclusion: out of the box, recognition is not as accurate.

So, with a throat mic you start with a lower accuracy but performance doesn't degrade with environmental noise. What this means is that hard recognition problems got harder (but you are insensitive to background noise). You can't do a lot of things with a throat mic. Taking the other perspective, if your problem is easy enough, throat mics work great (even in wind). The relevant question becomes, "how hard is our problem?"

The hard part of our problem is not the recognition of the valid commands, it is the rejection of random speech; it is the separation of command utterances from random speech. My first

use of the throat mic in the application was disappointing. I didn't make quantitative measures of its accuracy, but the 'feel' of its use was poor.

To begin a systematic study of the throat mic accuracy, I collected about 300 utterances that were 5 digit strings (zip codes). The initial throat mic system had an 18% to 20% string error rate on this test set. My signal processing was what I use for a conventional mic.

There was a straightforward way to optimize for throat mic data – a fairly simple change to my signal processing. I implement it.

The signal processing takes a window of digitized sound and boils it down to a “speech vector”. The scoring of a phonetic model is the computation of how the speech vector deviates from the model. The individual features of the “speech vector” are compared to corresponding part of the model, and the deviations are accumulated for the total score. A simple optimization would be to change how the individual features are weighted in contributing to the final score. I did this type of optimization.

After the optimization process the throat mic system had a 4% to 5% digit string error rate – a big improvement over the initial 18% (For comparison, using a conventional mic on the same task has less than a .1% error rate.) Fortunately, recognizing the command phrases of the recorder is easier than digit recognition. After this optimization the recorder application was very useable. So the combination of a throat mic with throat mic specific signal processing and that we are doing a simple recognition problem, enabled a recorder application that works in all environments.

As and aside, a person listening to throat mic recordings finds them very understandable. They sound somewhat muffled. A human listener quickly gets use to how they sound and has no issue with intelligibility.

There is one final development that was important in the success of the recorder application.

Confidence Measures in Enrollment:

The HandHeld Speech recognizer is speaker-dependent. This is both an asset and a deficit. Because it is speaker-dependent, after a person enrolls, it will work for everyone. The system does not care about accents or unusual voices, it learns how you speak. It creates custom models for every voice.

The downside is that you must enroll. The user must read prompt after prompt. The system has to study how you speak, then create custom models of your voice. Enrollment takes time and is monotonous. Even though they need only enroll once, users overwhelmingly prefer no enrollment. From the first version of the recorder application onward, a consistent criticism has been this need for enrollment.

The solution is to make the enrollment process require very few utterances, so few utterances that is no longer feels like enrollment. The solution is to pair speaker search with an utterance fit measurement.

The HandHeld Speech recognition engine is unique in its architecture. It is a Multi-Voice recognizer. It can run parallel competing recognizers. If we have a catalog of voice models

when someone reads a prompt, we can run recognition with all of them competing and discover the best fitting models. We do model search. If we have a broad selection of models, when someone enrolls, we can find an existing model set that is a close match to his voice. We find the best match of the existing models.

There is a caution here... we find the best fitting models, we don't actually know how good the fit is. This is where a fit confidence measure comes in. It was hoped that with the confidence measure, we could distinguish a few gradations in the fit. We hoped to realize when the fit was very good and that there was no need for further prompts (that the user had "found" his models). We hoped to separate from that, the case when the fit was only alright – when there was a need for a few additional prompts. And finally we hoped to be able to distinguish when the speaker had a very unusual voice (different from all in the catalog) and force an extended enrollment.

The hope was that with a large enough catalog most people would have virtually no enrollment.

This strategy was found to work. We are able to do all these things.

There are many experiments to perform to explore this use of the fit measurement. We have done very few of them. Under the time pressure of getting something working in the recorder application, we simplified the problem. We didn't need a global perspective on these fit measures work for our enrollment purpose, we need a detailed understanding of how the fit measures work with a particular prompt. We needed a narrow focus on one prompt.

We would like a nice long prompt with a rich set of phonetic data. With such, we would obtain a representative sample of the voice and a characteristic measure of how models fit the voice. Toward this goal, the longer the utterance the better. The difficulty of a long prompt is that it is easy for a user to mis-say it. We chose the prompt "1 2 3 4 5 6 7 8 9 10". It is long and because it is familiar, chances are very good the user will say it correctly.

The first thing we looked at was, what were the measures on this counting prompt for a good speaker using good (his own) models. How repeatable were those numbers, what was the normal variation? Then how did that change when the speaker spoke quickly or slowly.

Of the two fit measurements (x, y), x = the average penalty per frame was the more informative. y = the duration weighted penalty had small variation, it only gave a big signature when the person misspoke. The duration-weighted penalty would be useful to guarantee that they had actually said the prompt. We based the decision on the need for further prompts solely on the average penalty per frame measure.

A measurement on the counting prompt for good fitting models was typically 460 +/- 30. There was no significant difference when the pace of speech was changed.

The next question is, how do the numbers change when the speaker was using clearly "wrong models"? That answer was typically 560 to 600. So there is a clear difference between the expected range of measure when the models fit well and the values of badly fitting models.

These results were for a conventional mic. When the experiments were repeated with throat mics, all the numbers were systematically 10% lower. Again, there was a clear distinction between the normal variation of properly fitting models and ill-fitting models.

Good systematic experiments were not done (mapping this area out). Data came from 4 individuals where measurements were made with their correct models and then by forcing measures using the models of the other individuals. Because of time constraints, once I had

a feel for the values on the counting prompt I directly implemented a criteria in the recording application.

I made a cutoff of 450 for throat mic data, and 510 for a conventional mic. Cognizant of the fact that I was extrapolating from small data sets, I was cautious in my enrollment strategy. I still required the user to say some prompts, even when the fit measure was good. If you scored better than the cutoff, you were prompted to read another 6 prompts. If you scored worse, you were required to read approximately 60 prompts (the original enrollment process).

It is possible that we could extract the particular information about which phonemes have bad fits, and use that information to select future prompts. But that is future work.

In summary, after the search over models, if the models found are a good fit, enrollment can be exceptionally short. If the catalog of models were complete, everyone would experience short enrollment. The next logical question is how big is the catalog that covers most people?

A Catalog of Models:

There is nothing special about one set of voice models over another. They are all speaker-dependent models – just a description of phonemes for a particular speaker's voice. They can be well adapted or poorly adapted. They may have seen data on all of a speaker's phoneme models or they may have only been exposed to a small number of utterances (and only a small subset of models reflect the speaker's voice).

For a catalog, you would like the speaker's models to be “well adapted”. The question is, what does that mean? Let me clarify.

There are roughly 50 phonemes in English. For example there is a long 'E' sound, an 's' sound, and an 'l' sound, in both the words “lease” and in “seal”. Concentrating on the long 'E' sound, in the word 'lease', you are leaving an 'l' sound when you start the 'E'. Your mouth smoothly move into the 'E' sound from the 'l' sound. And later at the end of the 'E' before it is quite over, your jaw closes and you move your mouth toward the 's'. We represent that E as 'lE_s'. Similarly the E in seal is 'sE_l'. When we look at speech data it is clear that we will need different models for the different contextual E's. When there are 50 phonemes, there are $50^3 = 625000$ triphones. Some triphones are similar to each other and a model can be shared, but a lot aren't. There are a lot of models in our voice models. In a small vocabulary command situation, only a fraction of them are used. To be well adapted on a small problem it is not necessary to have all the models well adapted, only the ones you will use.

When assembling sets of models into a reference catalog, there are a few things to consider. If the catalog will be used on all problems, then all models need to be adapted. If there is only one target application, then only the phoneme models used in that application's command set need be adapted.

For the purpose of the recorder app, I opted to use models that were only well adapted on the small problem. I used the models where people had enrolled, then trained their models until they worked flawlessly for them in the application.

I had 5 sets of models, 4 male voices and 1 female voice. Two of the male voices were very similar, probably one could have been omitted but I didn't have data to tell which set was better, so I kept both. In the recognizer, I use a measure of the voice that corresponds to a pitch measurement, a vocal tract size measurement. Typical male voice pitch measures range from 50 to 60, female voices from 60 thru low 70s, children's voices into the low 80s. Using this measure, the voices in my catalog were:

52 – deepest male voice

55 – first of 2 similar, male

56 – second of 2 similar, male

58 – highest male in this set

62 – low female voice

None of the speakers had appreciable regional accents.

In the test situation (March 12, 2011) we enrolled speakers using the above 5 sets of voice models as our catalog. It turned out all of the speakers had good fit measures and got short enrollment. There were two unusual voices among them. There was a male speaker with a British accent, and a fairly high-pitched female speaker (pitch measure 66). Both got short enrollment. This was a surprise. I would have thought that none of the models were close enough to “work”. That suggests that a fairly coarse covering on the space of all voices will be sufficient. It may not take a large number of models to guarantee very short enrollments. There are future experiments to run to verify this suggestion.

There is one other observation to support this conjecture... When listening to throat mic data, the recordings are muffled and high frequency data is attenuated. In some ways the recordings sound more like each other. Your ear/brain is a very good signal processing system. If a person observes a similarity, it is probably there in the data. The way the throat mic affects the signal may make it easier to get a covering set of models.

One last note, it is probable that with especially good scores, there is no need for further prompts. With future testing we could be more aggressive about stopping enrollment. With a big enough covering catalog of models, we would have a system that is very close to speaker-independent. Most users would enroll in very few prompts... maybe only one... count to 10 for us. This verges on speaker-independence. Enrollment becomes say a warm-up prompt.

GumStix:

As we approached the end of the project, we began a port of the application to embedded Linux and GumStix Overo hardware (see www.gumstix.com). The breadboard solution used a USB connected Starix radio (the Puma). The breadboard solution contained the same cpu, same OS, as the new hardware radio solution being developed by Starix. The breadboard solution was a development platform for the new Starix hardware. In terms of research, there was nothing new discovered in this port, but it is important to document that we did it. This port was an explicit point in our statement of work. We finished this port and demo'ed the breadboard prototype at the final testing, OD3.

A Viewer:

We also created an accompanying Viewer to inspect recordings. It is also important to document that we created it and it works. The viewer is an explicit point in our statement of work. But again it is unimportant from a research perspective. Its use is documented in the Appendix B User Manual.

The Final Test, OD3:

The Recorder:

In the month prior to the test date, I rethought the whole application and created a new version different from previous ones. For the first time, I discarded all commands except turning the mic on and off. Unfortunately, I also introduced other changes. I introduced a more strict criteria for accepting a "WakeUp" command.

The testing exercise took place on March 12th, 13th, and 14th 2011. The first day was an enrollment/training day. There were 5 users enrolled. As reported above, 2 had unusual voices. The 13th was the first day of testing. It was only a partial success. Users had some frustration. About half the time the recorder would not turn on the first time they said "WakeUp WakeUp WakeUp", they would have to repeat it. Despite this, all users said they preferred this free form user interface (turn recorder on, say whatever you want, turn recorder off). (There was another system by Think-A-Move (another speech vendor) that had a more constrained UI with lots of commands to do particular things. Implicitly users were comparing the two User Interfaces.)

That evening, I fixed the application, I made the condition for accepting a WakeUp command less stringent. The next day I watched very carefully. After a few uses of the "Train Last Command" command, the use became flawless. It both accepted every proper command and it ignored all utterances that were not commands. There were 2 users that tested it, both gave positive reports. Both reported that after a bit of training, it worked flawlessly.

I then took the system back to the medic who had had difficulties the first day. He played with the new system and he agreed it was fixed.

There was one remaining criticism. They thought the choice of words for the "Go To Sleep" command must change. What if a patient with a head wound overhears the command and believes it is spoken to him? That is not a good idea, and a confusion like this should be avoided.

The Gumstix:

On the first day of OD3, March 11th, we demonstrated the gumstix breadboard port. We turned the mic on, made some recordings and turned the mic off. The files were transferred wirelessly to an EIC via the Starix Puma radio. We then listened to the recordings, verifying everything was working. It would be a small additional port to take the existing breadboard system and move it to Starix's new hardware.

Key Research Accomplishments:

By the final day of OD3, we had an application that worked well and was preferred by all users.

There were 4 discrete pieces of research contributing to its success. They are:

- 1) Discovery of the “right” User Interface for medic documentation – It is the total focus the medic applies while he is giving patient care constraining the functionality of the documentation process. Documentation cannot be more complicated than a simple recorder. The medic does not have the bandwidth to deal with something more complicated. Should he be required to do something more, patient care will suffer.
- 2) Optimization of recognition to throat mic data (the main solution to wind) – We had never encountered the problem of wind in all our previous work. We have encountered high noise in industrial settings, but indoors. Wind is a different animal, the turbulent noise occurs directly above the active mic element. There may be methods of filtering to mitigate its impact, but it is a hard research problem to develop it. The throat mic is a good solution. Optimizing the signal processing for throat mic data gave us a significant improvement in accuracy (errors: 18% → 4% on a digit recognition problem). The improved accuracy was crucial to its use.
- 3) Confidence measures used with a garbage rule for command rejection – This has been our first implementation of layering a fit measurement/confidence measure on top of a garbage rule. Rejection is improved an additional order magnitude above the use of a garbage rule alone. The combination lowers errors in command rejection below user irritation thresholds.
- 4) Confidence measure coupled with speaker search, minimize enrollment – Combining speaker search over a catalog of models with a fit/confidence measure, enables extremely minimal enrollment. Because the length of enrollment is tailored to each situation, a particular accuracy level in later use can be guaranteed. And given a good covering set of models, almost all users will spend almost no time enrolling. Speaker-dependent recognition becomes more attractive than speaker-independent.

Reportable Outcomes:

In this contract we built a prototype solution using many new techniques. They were all successful. The prototype itself is a reportable outcome. It has been nominated for MRMC 2011 MAMP to further develop it. Other reportable outcomes are in the planning stage. The problem of transcription is still very much a research topic. We plan on proposing a project to study and solve it. We also plan on creating a simplified civilian version of the recorder for iPhone and Android.

Conclusion:

First, speaker search coupled with a fit/confidence measure is a new solution to enrollment. The enrollment process for speaker-dependent recognition becomes easy and short. Enrollment is no longer the gauntlet issue making speaker-independent modeling preferable. This coupled with the two facts

- 1) Speaker-dependent recognition can be guaranteed to work for everyone and
- 2) Speaker-dependent modeling should have a consistently higher accuracy

makes speaker-dependent modeling more attractive than speaker-independent. This is new.

Another important conclusion is that layering a confidence measure with a garbage rule on command and control tasks, improves rejection by an order of magnitude. That factor lowers the pain of dealing with false positive below a user's irritation threshold. It makes an application "work".

Another conclusion, throat mics are a viable solution for wind. Compared to a conventional mics in quiet, they are inherently less accurate. But, they do not degrade with environmental noise, in particular high wind has no affect. Conventional mics become unusable for speech recognition purpose in high winds. So if a recognition task is easy enough or it can be made easy, throat mics are a good solution for outdoor use.

And a final conclusion, the medic in the field does not have the mind share to use a complicated application for documentation. His mind is fully occupied concentrating on patient care. A simple recorder is the best solution for documentation.

References:

Hui Jiang (2005) "Confidence measures for speech recognition: A Survey". Speech Communication 45 (2005) 455–470

Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin (2003). "A Practical Guide to Support Vector Classification". Technical Report Department of Computer Science and Information Engineering, National Taiwan University.

Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

Greg Gadbois (2009) "Applying Multi-Voice Speech Recognition to AHLTA-Mobile" (the final report to an earlier contract by the author) Contract GS-35F-4522G; W81XWH-06-F-0303

Appendix A:

Details of the Fit / Confidence Measure Calculation:

Here are the details of how the (x, y) of the fit / confidence measure are calculated. We illustrate it in the context of an example utterance.

Suppose someone spoke, saying “once upon a time” and that this happened to be one of the recorded utterances that the “garbage rule” does not correctly reject. Suppose that the valid command “respirations one nine” is the top choice after recognition. The words “respirations one nine” has the phoneme sequences: . r e s p R A sh N s . w u n . n I n .
(the '.' correspond to optional silence)

Suppose the utterance is 2.6 seconds long. A frame of data corresponds to 20 milliseconds of digitized sound. The utterance is 130 frames long.

The recognition process discovers the best alignment of how the frames of sound fit the phoneme sequence. For example the first 8 (of the 130)frames might be attributed to silence, then the next 5 fit the 'r', the next 8 → 'e', etc. There is a best segmentation of how the frames of sound aligns with the phonetic spelling. Each frame of data is assigned to a phoneme.

That frame has a score of how well that frame fit the phoneme model it was assigned to. If the data fit perfectly, the score would be 0. A slight misfit would be a small positive number. The worse the fit, the bigger the number. Accumulating the scores across all the frames is the score the recognizer returns with a recognition result. The top choice has the smallest score, the next best has a somewhat bigger score, increasing further as you move down the choice list.

In calculating the details of the fit, we first get the frame alignment information. The next thing is, we omit all the frame information where the frames were assigned to silence. We are only interested in the frames that are assigned to real phonemes. It is very important to remove the silence frames. Typically there are a lot of them. How they happen to fit the silence model has no bearing on how well the voice matches the models. Because there are a lot of frames of silence in an utterance, if you don't remove them from your measures, they can dominate calculations.

The first fit measurement (x in our (x,y) pair) is just the average score per frame (of the non-silence frames). We sum the scores and divide by the number of frames.

The second fit measurement (y in our (x,y) pair) is a duration weighted score per frame (of non-silence frames). The rationale is that when wrong sounds are being forced into a particular transcription, there are unusual alignments. For example there might be areas where the sound fits a phoneme well, and later, 2 phonemes down, there is another area of good fit. In the middle there may be an unusual number of frames being forced into the intervening phoneme. Probably the scores are not good either. This is a characteristic misfit,

a large number of bad fitting frames are being crammed into one phoneme. In preparation of defining the measure, we introduce some quantities:

Given S_i the score of a particular non-silence frame and S_E an average expected value of a frame score, we define a threshold score S_{Ti} :

if $S_i - S_E > 0$ define $S_{Ti} = S_i - S_E$ else $S_{Ti} = 0$

So S_{Ti} is strictly positive and is 0 if the score is below some expected value.

Next we define a duration weight W_d . Suppose that on average in a typical good recognition, 8 frames are being assigned to each phoneme. We define an expected duration $D_E = 8$.

We define D_i to be the actual duration in the i_{th} phoneme.

If $D_i - D_E > 1$ define $W_d = D_i - D_E$ else $W_d = 1$

Then we find the average value of $W_d * S_{Ti}$. We accumulate the product of $W_d * S_{Ti}$ over the non-silence frames, then divide by the number of non-silence frames. This is our y , our duration weighted score per frame.

We can extract an (x, y) for any recognition result, it is a simple matter to keep running averages of them (X, Y) .

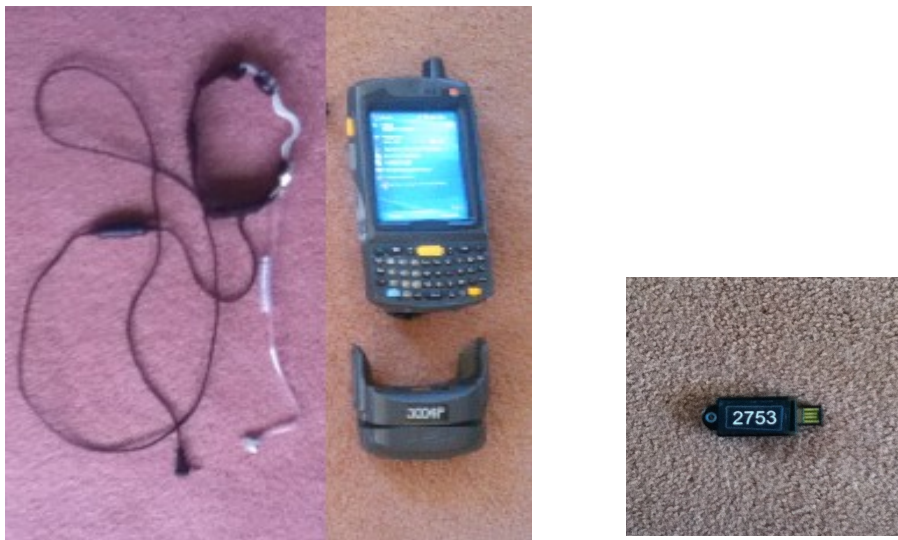
Appendix B:

Hands Free Voice Recorder User Manual

Overview:

The Hands Free Voice Recorder is a software solution for capturing more/better medic documentation from the field. The recorder is turned on and off using voice commands. Feed back is primarily oral, operation is eyes-free. The command set is kept purposely small so that it does not distract the medic from his primary mission.

The Hardware: The hardware for this version of the recorder consists of a Motorola MC70, a throat mic, and a short range radio provided by Starix. Each patient will have a memory stick device – the “EIC” (Electronic Information Carrier). The radio allows a wireless movement of recordings from the MC70 to the EIC. The hardware is pictured below:



Throat Mic

MC 70 and Radio

EIC

Getting Started:

The Starix radio is powered by a rechargeable battery. You may need to recharge it. There is a 5 volt DC charger included. Plug it in to the back side of the radio. It takes 3 to 4 hours to fully charge.

The EIC is also powered by a rechargeable battery. You may need to recharge it. The EIC charges when it is plugged into a USB port. The lights on the EIC will blink when it has good contacts and is charging. Again, full charge is reached after 3 to 4 hours.

Immediately after removing the EIC from the USB connector, you will see lights on the EIC blink either once or twice. If it blinks once, the EIC is in a “dormant” state, it will not “connect” to the radio. You must briefly plug it back into the USB slot and then remove it. This time it will blink twice. The EIC is now in its “active” state. Each time you plug it in and remove it, it switches between dormant and active (one blink or two).

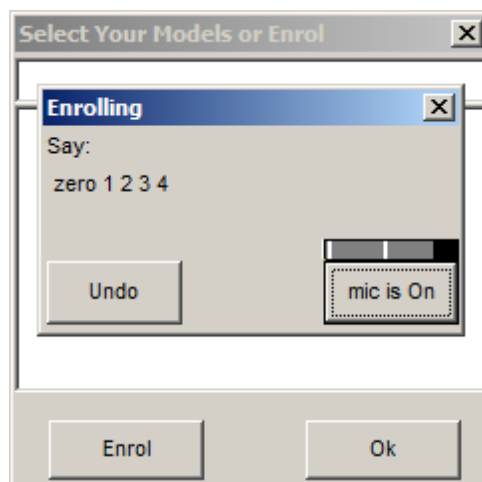
The hardware assemblies:

- 1) Slide the Starix radio onto the bottom of the MC70.
- 2) The mic jack plugs into the MC70. The hole for the jack is located on the right side of the MC70 approximately parallel to the top of the screen. (There is a little rubber flap covering the hole.)
- 3) The mic is worn like a choker. The clasp for the mic is two magnets that fit together.. Practice separating the magnets by holding one side in each hand and twisting your wrists in opposite directions. When attached, the clasp should be at the back of your neck. The clear rubber “nose piece” is centered on your trachea. The neck sizing is adjustable. The mic should fit snugly so there is pressure holding the vibration sensors against your neck. The pressure should be noticeable when you first put it on but it should rapidly be ignoreable (so that when you are focused on some task, it is completely unnoticed). If it is too tight, adjust it. It should not be choking you. It should have small pressure but be ignorable. The earpiece goes in your ear. If the earpiece is on “the wrong ear” you can take the mic off, flip it 180 degrees and put it back on.

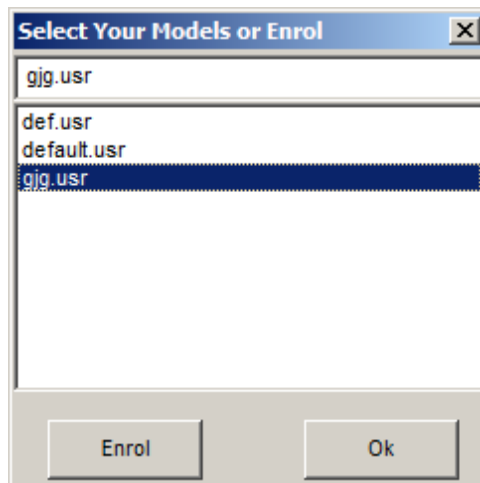
The application installs to an “\hhs” folder off of the root. In that folder is the executable “\hhs\rcrdTm.exe”.

Run that executable.

If this is the first time the application has been run, and there are no existing models, you will be required to enroll. You will be taken directly to the enrollment dialog, you will see the following screen:

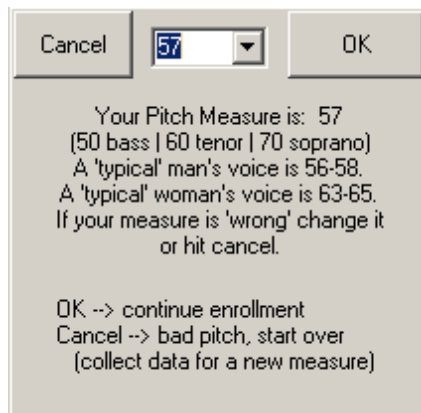


If there were existing models you could either select your models then tap “Ok” or you could choose to enroll.



As a first time user of the system, you will need to enroll, tap the “Enrol” button.

You will then see the Enrol dialog. You will be told to “repeat the prompts”, then you will hear the first prompt “zero 1 2 3 4”. Echo back “zero 1 2 3 4 “. If the system has heard your response, you will hear the next prompt, “5 6 7 8 9”. You say it back. You will then hear the third prompt “1 2 3 4 5 6 7 8 9”. Say it. The application then pauses, a measure of your voice is being calculated. It is extracting a measurement of how deep your voice is. If you have a very low deep male voice, the number may be as low as 50. A typical man's voice is in the mid thru upper 50's. A typical women's voice measures is in the 60's. After a small wait (it had to think a little bit, calculating the measure), you will see the following dialog with the measurement of your voice showing:



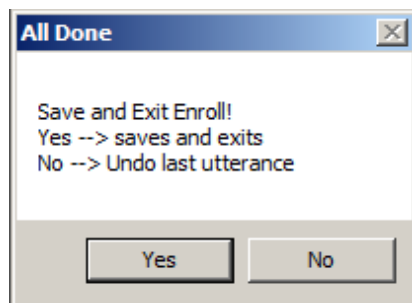
If you hit “Cancel”, you will be re-prompted with “1 2 3 4 5 6 7 8 9” and a new measurement made. Typically you will see the same number plus or minus one. The measure should be very reproducible. If you are satisfied with the number, tap “OK”.

Next you will be led thru a further series of prompts. Depending on the measurements made on the long counting prompt, you will either get a short set of prompts consisting of:

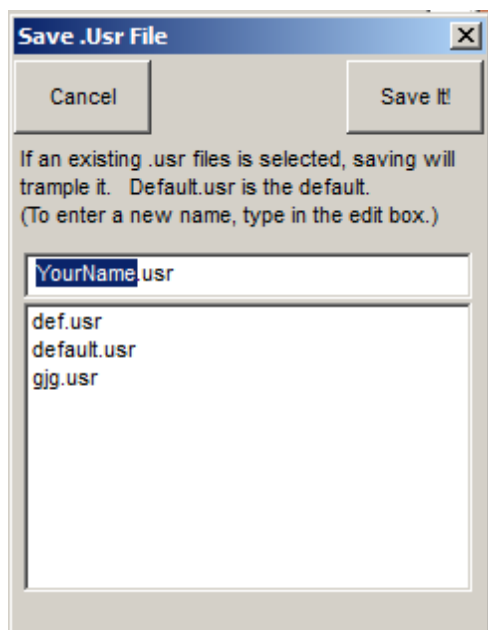
- 1) "go to sleep"
- 2) "wake up wake up new patient"
- 3) "medications"
- 4) "vital signs"
- 5) "interventions",
- 6) "progress report"
- 7) "patient summary"
- 8) "train last command"

or you will get a long series of about 60 prompts. Most people are only asked for the above 8. If you have a very unusual voice, you will be asked to repeat the long set. At any time during the enrollment, if you mis-speak, or you think something was not quite right, tap the undo button. You can back up the prompts and re-do them, as many as you need. One last note, you should speak the prompts in a relaxed flowing sort of way. You don't want to talk in a rigid discrete word way. The voice models it creates are based on how you speak to it during this enrollment. If you speak in a stilted fashion, it will expect you to speak in the same fashion during real use.

Once you have finished repeating prompts, you are asked to save and exit enrollment.



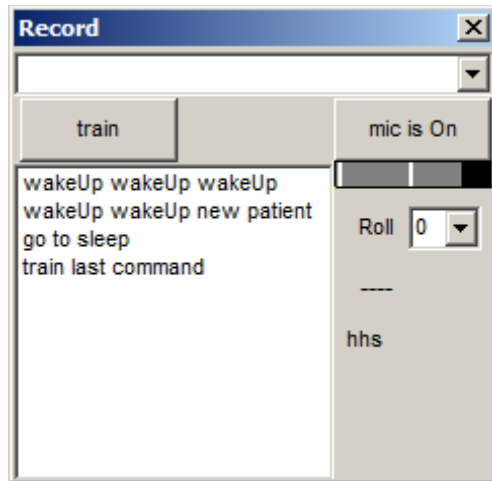
You tap yes, the program will be busy for a bit, while it makes your voice models, then the following dialog appears:



In the text box (where it currently say "YourName.usr") type your name. You can bring up the onscreen keyboard by tapping the icon at the bottom center of the screen. After entering your name, tap "Save It!". You have finished creating models.

The next dialog is the application's main screen. This is the Recorder.

The Recorder:



A user typically only enrolls once. After he has created model, he selects them at the opening screen and comes directly to this dialog.

The recording dialog is meant to be operated eyes and hands free.

There are only 4 commands. They are:

- 1) WakeUp WakeUp WakeUp
- 2) WakeUp WakeUp New Patient
- 3) Go To Sleep
- 4) Train Last Command

They are listed in the text box as a reminder. When commands are spoken, they must be spoken in isolation. That means there must be a slight pause before you say the command and another pause after it (before you say anything else).

There is an on screen mic button that will turn the mic on and off. Generally one never uses it. The mic is always on. It listens to everything you say and ignores it all, until you tell it to wake up.

There are 2 wake up commands. When first presented with a new patient or switching between patients, you say "WakeUp WakeUp New Patient". This command does two things, it turns recording on and it initiates a connection with the closest EIC. When recording data about a patient, the application needs to know where to store the recordings. When you say "WakeUp WakeUp New

Patient”, the MC70 scans (using the Starix radio) for the closest EIC and connects to it. The user gets confirmation of which EIC is found, he will hear the EIC id (the 4 digits printed on the EIC) read back to him. Subsequent utterances are then collected and stored both on the MC70 and the EIC. (The medic retains a local copy and a copy accompanies the injured soldier.) The silence between utterances is discarded. The medic is free to say anything he wants. When he says “Go To Sleep” recording is stopped and the recorder returns to the dormant state. Later he can either say “WakeUp WakeUp WakeUp” to use the previous EIC. Or he can say “WakeUp WakeUp New Patient” to scan for the closest EIC. While dormant (sleeping) the MC70 is actively listening to everything spoken and discarding the utterances (until it determines a wake up command is heard).

As an example scenario, suppose a medic has one injured soldier. He begins documentation by approaching the patient (so that the closest EIC is the one on the patient). He glances at the EIC and says:

“WakeUp WakeUp New Patient”

He hears the digits spoken back (they match the digits on the EIC). He then says:

“Gun shot wound to the right arm. Pressure dressing applied.”

He will hear a little ding. It is an acknowledgment that the utterance was captured, everything is operating correctly. He does not have to note the time, the application records the time and date automatically for each utterance. He is temporarily done. He says:

“Go to Sleep”

He hears a two toned “ding dong” confirming the recording is stopped. He speaks to the patient and measures blood pressure. (The application ignores the utterances, there are no beeps.) He says:

“WakeUp WakeUp WakeUp”

He hears the same digit string as before as confirmation that recording is restarted and the recordings will be recorded to the same EIC as before. (There was no new search for the closest EIC – ‘WakeUp WakeUp WakeUp’ implies use the last known EIC.) He says:

“Blood Pressure 140 over 70”

He hears the confirming ding, then says:

“Pulse 65”

hears another confirming ding, and finally says:

“Go to sleep”

He hears the “ding dong” confirming recording is stopped.

That is the typical usage. The user uses one of the 2 WakeUp commands to turn the recorder on. Then he says whatever he wants to say however he chooses to say it, and utterances are recorded. Finally, he turns recording off by saying “Go to sleep”. Turn it on, say what you want, turn it off. Usage is very simple and does not distract the operator from the primary job of giving care.

Documentation can happen even during hectic care giving.

There is one other oral command. Suppose the user said "WakeUp WakeUp New Patient" and nothing happens, there is no confirmation of the command. The application rejected the utterance, it did not believe it was a valid command. Suppose the user repeats the command, saying "WakeUp WakeUp New Patient" and this time the digit string of the closest EIC is repeated back. The user immediately says "Train Last Command". He will hear "trained" spoken back to him. What has happened is the last utterance which the application recognized as "WakeUp WakeUp New Patient" has been used to adapt the users voice models. The phonetic models that were created during enrollment had a description of what "WakeUp WakeUp New Patient" should sound like. The actual "WakeUp WakeUp New Patient" that was just heard is compared to the description and the description is adjusted to look more like the last utterance. The phonetic models are moved in the direction of the last utterance. This adjustment is done cautiously. The difference between the model and the actual utterance is measured and the models are move 10% of the way toward the utterance. It may take a few instances of training to make a serious change in the models. But correspondingly, if the operator says something a little wrong and forces training, the models won't immediately become unusable.

The "Train Last Command" command allows the user to selectively train his models. If the application is not working perfectly (the operator feels like he is fighting with it to recognize a particular command) he is empowered to "fix it". He can train his models on the fly with real utterances under real conditions. He can make the system work. This is an important feature of the system.

To summarize, the basic operation is to turn the recorder on, say what ever you want however you want to say it, then turn the recorder off. All interactions are verbal with both eyes and hands free. And there is audible feedback so that it is clear recordings are properly occurring. This is the user interface that the medic experiences.

In addition to this basic operation, word spotting / topic spotting is occurring on the collected utterances. The utterances are categorized as to whether they are information on
vital signs
medications
interventions
report information (wound type, location of injury, consciousness level, everything else)

All utterances are tagged with a time stamp, with the category of the utterance, and with the EIC id identifying the patient. We consider the recordings the primary data from the field. When only important utterance data is recorded and all intervening silence is removed, there may only be a minute of recordings. It is very easy to listen to all of it. These recordings are digital and can be pushed forward to the field hospital before the injured soldier arrives. It is not onerous to play all of them.

Trouble Shooting:

You say "Blood Pressure 140 over 70 Go To Sleep" and you got a confirming ding of recording, not the "ding dong" of turning the recorder off... You didn't pause, separating the Go To Sleep command from the previous utterance. Say commands in isolation.

You say "WakeUp WakeUp New Patient" and nothing happens.
Check all the plug on the microphone, unplug them and plug them back in.

Make sure the mic button says "mic is On".

Check that the VU meter under the "mic is On" button moves when you talk, is it moving?

Toggle the mic button on and off.

When you talk, do garbage words appear in the top combo box? If so open the combo box and check if "WakeUp WakeUp New Patient" is in the list. If it is, select it then hit the train button.

If the mic appears to be working and after multiple attempts you can't find "WakeUp WakeUp New Patient" in the choice list, re-enroll.

You say "WakeUp WakeUp New Patient" and you hear the wrong digit string.

The wrong EIC is closer. Get closer to the "right" one. Try again.

Possibly the EIC is in its dormant phase, it needs to be activated by being temporarily plugged into a USB port then extracted. When you pull the EIC out of the USB, it should blink twice.

Possibly the battery is dead in the EIC, re-charge it (plug it into a USB for 4 hours).

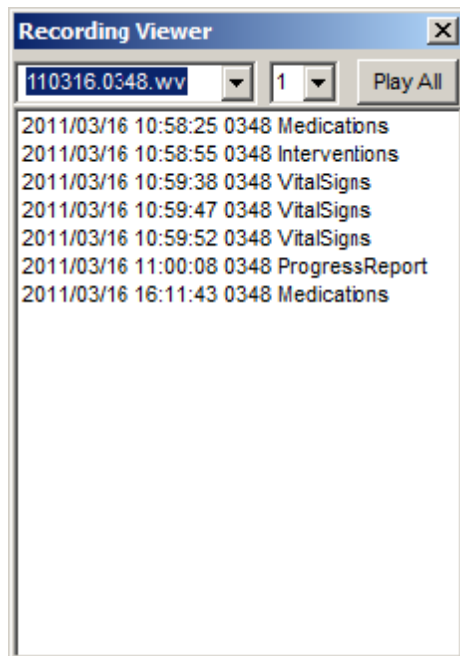
You say "WakeUp WakeUp New Patient" and you hear "No active EIC". Probably the radio on the bottom of the MC70 has been banged and was temporarily disconnected, causing the radio to reboot. Shut the application down and restart. (Don't worry, no data will be lost. The complete recordings are on the local MC70, the next time you reconnect to the EIC and add a recording, all back recordings will be added.) If after relaunching problem persists, the radio needs to be recharged.

Finally there is the recordings.

The Recordings:

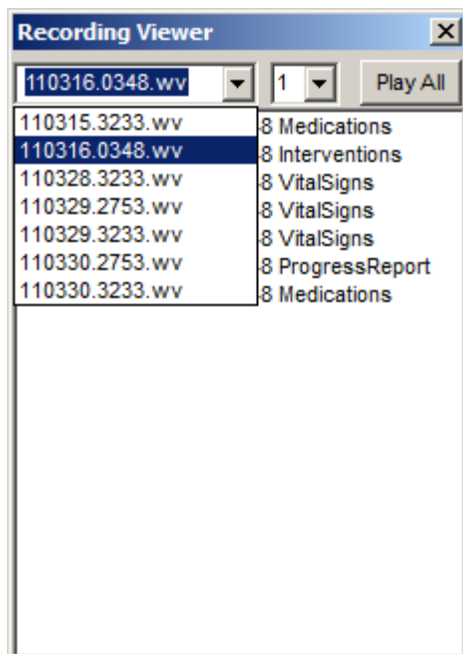
When you look on the MC70, on the "Storage Card" the folder that is the 1Gig SD Card, you will find files with names like, "110316.0348.wv" those are the recordings. Then first six digits correspond to the date. 11 → 2011, 03 → March, 16 → the day of the month, 110316 → March 16th 2011. The next 4 digits correspond to the EIC id. The same file will exist on EIC "0348".

11031.0348.wv is a compound file. It is really a bunch of digitized .wav files appended one after another. We ship a viewer to look and listen to these files. There is an application for both the desktop computer and for the MC70 called "RcrdView.exe". If you put the 11031.0348.wv in the same directory as the viewer, then launch RcrdView, you will see:



Each line in the list corresponds to a recorded utterance. The first one occurred on March 16th 2011 at 10:58. It was recorded while attached to EIC 0348 and the recognizer believe it to be a statement about a medication. If you were to tap that line, you would hear "2.5 milligrams morphine". You could tap the next line to hear it. Or you can press the down arrow key to scroll to the line below. By continually pressing the down arrow key you will hear them all.

The combo box in the top left, shows the name of the .wv file we are looking at. If we open the combo box we can see all the .wv files in the directory. They are sorted in alphabetical, chronological order. We can select another one to view.



In conclusion, we would note that it is very easy to listen to all the utterances. When the silence between utterances is discarded and only important utterances are recorded, the duration of the recordings is short, for example the 7 utterances in the 11010.0348.wv file was in total 15 seconds long.

At this time, it is not clear how or where these recordings would be stored with medical records. In addition to this viewer, there are tools to extract regular .wav files out of compound .wv files. Also there is a software toolkit so that it is easy to build custom extraction tools.

For further questions or development, contact HandHeld Speech at:

Greg Gadbois
greg@handHeldSpeech.com
978 388-0396