BlueMsg: Hands-Free Text Messaging System

Wireless Communication Systems

by

Sahir Ali
Savita Shetty

under the guidance of

Prof. Christopher Rose

RUTGERS
NEW BRUNSWICK

Department of Electrical Engineering
May 2009

Abstract

This project will develop a hands-free system of communication, between a person and a phone. This system will incorporate wireless technology to synchronize with a regular smart phone to generate a text message without the use of fingers, enabling this activity in a car or in a mobile environment amicably. The hands-free system will be used to integrate speech to text algorithms, from concepts of digital signal processing, and wireless communications to compose a text message by command of voice; providing a option of listening to the composed message before the message is sent automatically. The completion of such a project will provide safer means of communicating in a car which will be a step beyond the production of hands-free Bluetooth for phone calls.

Contents

List of Figures

1 Introduction

With the evolving of technology, we have been able to accommodate our short comings in multi tasking and prevail through hands-free systems. This system enables us to be productive and sometimes less destructive. Hands-free has allowed man to engage in a phone call without having to hold the phone up with hand, preventing hands off the wheel in a car. However, this system has not been developed enough to solve other such limitations, as composing a text message, a more lethal activity to incorporate while driving. The fruits of DSP (digital signal processing) in speech to text conversion and wireless communications can be integrated to generate this basic solution as an implementation of bluetooth hands-free technology. The success of such a system would modify such a dangerous synchronization of two tasks into a feasible and efficient possibility and carry the growth of technology beyond the recent trend of luxury to a primal necessity.

1.1 Motivation

We drove our motivation from the simple fact that cell phone usage while driving poses a great danger. According to the research at Harvard[1], 1 out of 20 car accidents are caused by cell phones. Moreover, there are a grave percentage of people who send out SMS while driving.

- Ages $18 - 24 \longrightarrow 50\%$ [2]
- Ages $25 - 34 \longrightarrow 30\%$
- Ages $35 - 44 \longrightarrow 19\%$

Recent technological advances in state-of-the-art wireless communication have harbored improved and efficient hands-free system to receive phone calls. However, these commercial hands-free systems are not self-suficient; they do not provide a solution for a hands-free text messaging.

1.2 Proposed Idea

A hands-free wireless text messaging system which will allow for the dictation of text message via voice and convert it to text for automatic transmission. It will be geared to be utilized in-car and PC (Personal Computers).

1.2.1 Wireless Communication Components

1. SMS
2. Wireless GSM Modem

Details of each of these technologies are explained later on.

[1] CBS News: Dec 2002 http://www.cbsnew.com/stories/2002/12/02/tech/main531320.shmtl

[2] Website:Edgar Snydar & Associates Date:Sept 2008 Article name: Almost Half of Young Drivers Text While Driving http://www.edgarsnyder.com/news/auto-accident/young-drivers-text.html?ref=http%3A//www.google.com/search%3Fhl%3Den%26sa%3DX%26oi%3Dspell%26resnum%3D0%26ct%3Dresult%26cd%3D1%26q%3Daccidents+caused+by+cell+phones+edgar+snyder%26spell%3D1

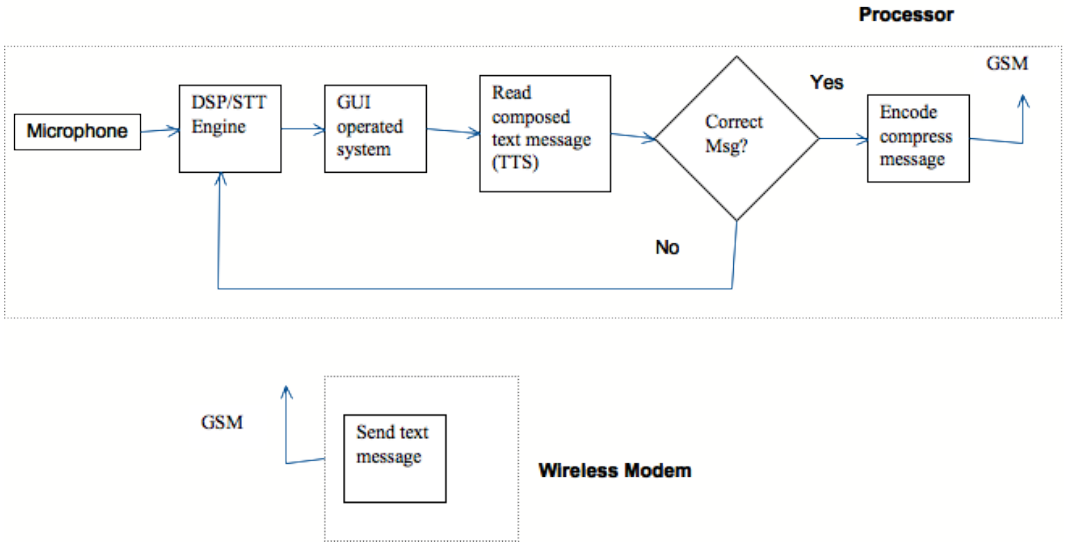


Figure 1: System Block Diagram

2 System Overview

Figure 1 has the block diagram of the entire system. User's speech will be picked up via microphone and will be processed by the signal processing block for conversion to recognizable text format. After the text conversion, data will be passed to the machine learning block, which will correlate words with some common words which are normally used in abbreviated form. Converted text message will be read back to the user for confirmation. From there on, the information will be modulated and sent via wireless GSM Modem.

Another adoption of this system: Cell phone will receive the information from the computer via bluetooth and will pass the information to the custom software for the transmission of the text message via service provider (SMS).

3 Overview of Methodologies And Technologies

3.1 DSP - Speech To Text

Speech recognition (SR) systems allow people to control a computer by speaking to it through a microphone, either entering text, or issuing commands to the computer. Early systems used discrete speech, i.e. the user had to speak one word at a time, with a short pause between words. Over the past few years most systems have used continuous speech, allowing the user to speak in a more natural way[3]. Microsoft have included their own speech recognition system within recent versions of Windows and have made the API available for programmers. Speech-to-Text(STT) synthesis is an essential part of this system as it takes the users voice and converts it to text for further processing and wireless transmission.

3.2 DSP - Text to Speech

The goal of Text-to-Speech (TTS) synthesis is to convert arbitrary input text to intelligible and natural sounding speech so as to transmit information from a machine to a person[4]. This particular technology will be used to read out the converted text for user's confirmation

3.3 SMS (Short Message Service)

SMS stands for Short Message Service. It is a technology that enables the sending and receiving of messages between mobile phones. SMS first appeared in Europe in 1992. It was included in the GSM (Global System for Mobile Communications) standards right at the beginning. Later it was ported to wireless technologies like CDMA and TDMA. The GSM and SMS standards were originally developed by ETSI. ETSI is the abbreviation for European Telecommunications Standards Institute. Now the 3GPP (Third Generation Partnership Project) is responsible for the development and maintenance of the GSM and SMS standards.
As suggested by the name "Short Message Service", the data that can be held by an SMS message is very limited. One SMS message can contain at most 140 bytes (1120 bits) of data, so one SMS message can contain up to:

- 160 characters if 7-bit character encoding is used. (7-bit character encoding is suitable for encoding Latin characters like English alphabets.)
- 70 characters if 16-bit Unicode UCS2 character encoding is used. (SMS text messages containing non-Latin characters like Chinese characters should use 16-bit character encoding.)

SMS text messaging supports languages internationally. It works fine with all languages supported by Unicode, including Arabic, Chinese, Japanese and Korean.
Besides text, SMS messages can also carry binary data. It is possible to send ring-tones, pictures, operator logos, wallpapers, animations, business cards (e.g. VCards) and WAP configurations to a mobile phone with SMS messages.

One major advantage of SMS is that it is supported by 100% GSM mobile phones. Almost all subscription plans provided by wireless carriers include inexpensive SMS messaging service. Unlike SMS, mobile technologies such as WAP and mobile Java are not supported on many old mobile phone models.

3.4 Wireless Communication - GSM Modem

A GSM modem is a wireless modem that works with a GSM wireless network. A wireless modem behaves like a dial-up modem. The main difference between them is that a dial-up modem sends and receives data through a fixed telephone line while a wireless modem sends and receives data through radio waves.

A GSM modem can be an external device or a PC Card / PCMCIA Card. Typically, an external GSM modem is connected to a computer through a serial cable or a USB cable. A GSM modem in the form of a PC Card / PCMCIA Card is designed for use with a laptop computer. It should be inserted into one of the PC Card / PCMCIA Card slots of a laptop computer.
Like a GSM mobile phone, a GSM modem requires a SIM card from a wireless carrier in order to operate.
Computers use AT commands to control modems. Both GSM modems and dial-up modems support a common set of standard AT commands. You can use a GSM modem just like a dial-up modem.
In addition to the standard AT commands, GSM modems support an extended set of AT commands. These extended AT commands are defined in the GSM standards. With the extended AT commands, you can do things like:

- Reading, writing and deleting SMS messages.
- Sending SMS messages.
- Monitoring the signal strength.
- Monitoring the charging status and charge level of the battery.
- Reading, writing and searching phone book entries.

The number of SMS messages that can be processed by a GSM modem per minute is very low – only about six to ten SMS messages per minute.

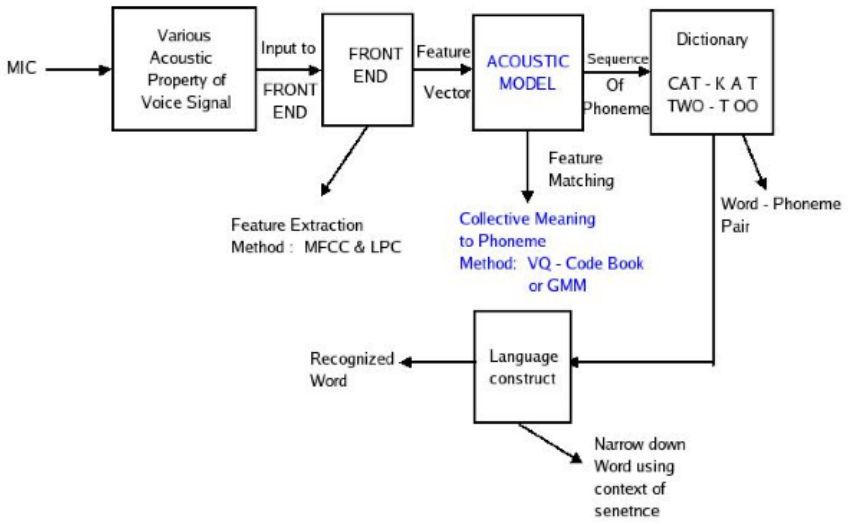


Figure 2: Block Diagram: Speech Recognition using HMM

4 Speech Recognition using Hidden Markov Model (HMM)

A block diagram of the speech recognition is given in fig

4.1 Feature Extraction

This is the front end processor for the speech recognizer. This module extracts the key features from speech. Feature extraction involves the following.

- **Word Boundary Detection**
  This is an important part of the feature extraction block. We have to isolate the word utterance from the starting and trailing noise. This was done by using Energy Threshold comparison method. Whenever, the energy in a frame of speech exceeds a certain threshold, we can mark this point as the start of speech. The same process can be repeated from the end of speech sample to detect the end of speech.

- **Pre emphasis**
  The digitized (sampled) speech signal s(n) is put through a low order digital system to spectrally flatten the signal. The first order filter used had the transfer function
  $$H(z) = 1 - az^{-1} \tag{1}$$
  where a = 0.9

- **Frame Blocking**
  The pre emphasized speech is then blocked into frames by using Hamming windows. Hamming windows of length 256 was used. To have a smooth estimate we need more windows. So, an overlap of 156 samples was also incorporated.

The hamming window used was

$$w(n) = 0.54 - 0.46cos(2\pi\frac{n}{N-1}) \tag{2}$$

- **Cepstral Coefficients Extraction**
  This is the crux of the feature extraction block. Cepstral coefficients can be used as features. Cepstral coefficients are the coefficients of the fourier transform representation of the log magnitude spectrum. These are more robust and reliable than the LPC coefficients. The cepstral coefficients can be estimated from the LPC.

- **Parameter Weighting**
  Low order cepstral coefficients are sensitive to overall spectral slope and higher order spectral coefficients are sensitive to noise. So, it has become a standard technique to weight the cepstral coefficients by a tapered window so as to minimize the sensitivities.

- **Temporal Cepstral Derivative**
  The cepstral coefficients provide a good representation of the local spectral properties of the framed speech. But, it is well known that a large amount of information resides in the transitions from one segment of speech to another.

4.2 Vector Quantization

The results of the feature extraction are a series of vectors characteristic of the time-varying spectral properties of the speech signal. These vectors are 24 dimensional and are continuous. We can map them to discrete vectors by quantizing them. However, as we are quantizing vectors this is Vector Quantization. VQ is potentially an extremely efficient representation of spectral information in the speech signal. The key advantages of VQ are :

- Reduced storage for spectral analysis information
- Reduced computation for determining similarity of spectral analysis vectors. In speech recognition, a major component of the computation is the determination of spectral similarity between a pair of vectors. Based on the VQ representation this is often reduced to a table lookup of similarities between pairs of codebook vectors.
- Discrete representation of speech sounds

4.3 Hidden Markov Model (HMM)

Hidden Markov Model (HMM) is a mathematical model for a set of states, generally multidimensional, associated with a probability distribution following the Markov process. Markov process defines a memoryless system where probability of the future state is dependent on the present state, or conditional probability illustrated in fig 3. However, in HMM the states are "hidden" but can be associated with the observation

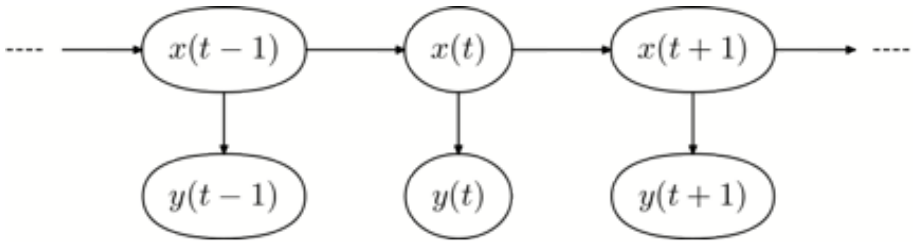


Figure 3: Markov Property States

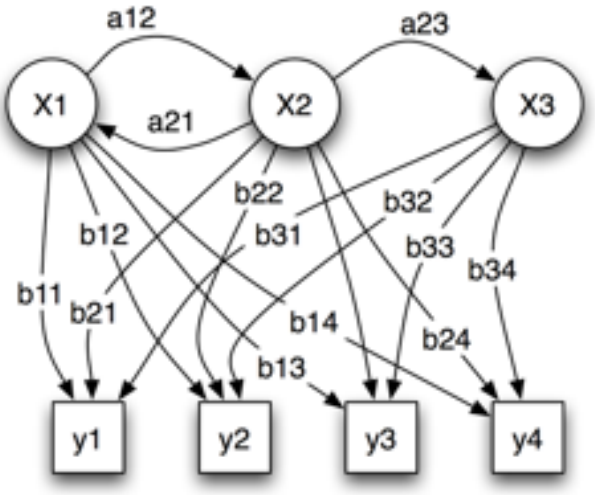


Figure 4: Hidden Markov Model schematic

or output states which they generate. A simple pictorial description can be seen in fig. 4, where the 'x' is represented by the various states (hidden in this model), 'a' is called the transition probability or the probability of one state leading to another, 'a' is called the emission probability or the probability a particular state can generate a particular output, and 'y' is represented by the observation or output states. Hidden Markov Model is used in various temporal pattern recognition such as speech, handwriting and gesture recognition.

4.3.1 CONNECTION TO PROJECT

For the purpose of this project in conjunction with HMM, the hidden states are words stored in the speech engine dictionary, which the transition probability accounts for the usage of specific words in specific orders in a sentence. This part of the speech recognition related to HMM is apparent. The observation stated or sequence is represented by the spoken words needed to be recognized, while the emission probabilities are modeled in the learning part of the system.

4.3.2 BASIC EXPLANATION OF PROJECT PARAMETERS

In the speech recognition system, the states and observations are stored in different forms. The observation, which is the speech input, is measured as an acoustic signal. The dictionary has a text representation of this acoustic sound which corresponds to the states in consideration. The probability of concatenating specific words in a sentence holds the key to the recognition process. For example the words "cat driver height mat" or some other ridiculous ensemble of words in a sentence would carry little to no probability in the trained dictionary being used by the recognition engine. However, a more common or sensible sentence holds a high probability and can be one of the states which gives rise to this observation. A mathematical structure of this system would help in the understanding of the recognition process.

- Let N be the number of states in the speech model (words in the dictionary)
- Let M be the observational states (words in the speech input)
- Let a_ij represent the transition probability between two states such that: $a_ij = p(q_{t+1} = j|q_t = i)$ where $1 <= i, j <= N$, q_t defines the current state, and $a_ij >= 0$, $\sum_{j=1}^{N} = 1a_{ij} = 1$
- Let $b_j(k)$ be the probability density of sentences constructed using the states: $b_j(k) = p(o_t = v_k|q_t = j)$ where $1 <= k <= Mand1 <= j <= N, b_j(k) >= 0$ and, $\sum_{k=1}^{M} b_j(k) = 1$

Furthermore, there are three stages which are involved in the calculation and recognition of speech input. The first is the evaluation stage which determines whether the input in valid. The second is the decoding stage which recognizes the speech input. The third stage is the learning stage which enhances the dictionary and personalizes the recognition system. A clear explanation of the three stages are presented below.

4.3.3 DETAIL OF EACH STAGE

UNDERSTANDING THE OBSERVATION (FORWARD ALG) The preliminary step at hand is to judge whether the received input in valid based on the current information from the dictionary. This process checks to see if spoken words or something similar is stored in the speech dictionary being used. Mathematically this process calculates the conditional probability of the observed speech input given the probability of the stored data or P(O|D) where 'O' is the observed speech and 'D' is the data in the dictionary. This calculation is carried out by the forward algorithm. The algorithm is relatively low in complexity as it uses a recursive approach to calculate each acoustic input received to the modeled information. The usage of this stage is necessary in the transition of the next two stages. Once the probability is determined, given by equation 1, a list of state sequences are generated which could match to the speech input. However, if no such sequence is found then the third stage

kicks in and learns the new vocabulary. To elaborate further on this algorithm, the following equations are presented.

$$\alpha_t(i) = p\{\theta = 1, \theta_2, , \theta_t, q_t = i|\lambda\} \tag{3}$$

where ? represents the input words up to the current time t in the calculation, and ? represents the model of the stored data in terms of a_{ij} and $b_j(k)$ as mentioned above. Using a recursive method, the following calculation holds: $\alpha_{t+1}(j) = b_j(\alpha_{t+1}) \sum_{i=1}^{M} \alpha_t(i) * a_{ij} where 1 <= j <= N, 1 <= t <= T-1$ (T is the time of the final received input word), and$\alpha_1(j) = b_j(\theta_1)$ The probability of the input being one of the stored states:

$$P(\theta|\lambda) = \sum_{i=1}^{M} \alpha_T(i) \tag{4}$$

SEARCHING THROUGH SPEECH DIC FOR LIKELY STATES (VERTIBI ALG) The previous stage works in accordance with the decoding or recognition stage. The search for the existence or the speech input in the current dictionary assembles a list of possibilities (this is the yellow text which forms at the bottom of the recognition window). However, one of the sentence structures from the list is most likely the spoken input. The Viterbi algorithm calculates the sequence of words with the highest probability match to the spoken words by using a recursion algorithm most commonly used in graph theory. The first piece of the input is matched to a word in the dictionary and the rest of the sentence is decoded by recursively calculating the highest transition probability to another word in the dictionary. If this word doesnÕt match the next observation input then the algorithm moves to the next best word in the transition path. The algorithm keeps a pointer to the highest matched probability between the observation sequence and stored data sequence, j*. The final sequence or sentence is printed based on the pointer. The mathematical description of the algorithm is as follows. $\delta_t(i) = max_{q_1,q_2,,q_{t-1}} p\{q1, q2, , qt = i, \theta_1, \theta_2, , \theta_{t-1}|\lambda\}$, which looks for the highest probability of the recognized states to the observed input up till the current time t. The follow recursive step in used to carry out this calculation: $\delta_{t+1}(j) = b_j(\theta_{t+1})[max_{1<=i<=N}(\delta_t(i) * a_{ij})]$ where $1 <= i <= N, 1 <= t <= T-1$, and $\delta_1(j) = b_j(\theta_1)$ The final recognized sentence is given as:

$$J* = arg[max_{1<=j<=N}(\delta_T(j))] \tag{5}$$

4.3.4 LEARNING NEW VOCAB (MOST LIKLIHOOD)

This stage occurs when the initial evaluation stage fails to generate possible results. The operation of this stage relies on the maximum likelihood criterion. The system tries to maximize the probability of the newly observed sentence with respect to the already existing sentence in the dictionary. Every time a specific sentence is spoken the probabilistic weights allotted to the transition between each word is increased relative to the other sequences in the model. The system effectively fits a new phrase into the speech dictionary and adds importance to the most frequently used phrases

keeping the speech system personalized to particular users or tasks.

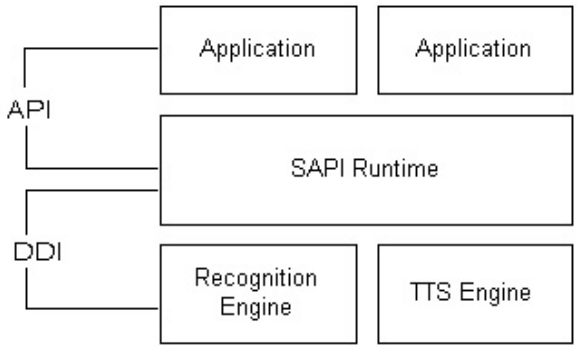


Figure 5: Block Diagram: High-level view of SAPI

5 BlueMsg Software Design

In this section, we will describe the engineering behind our software. Our software has two main cores: Speech Processing and Text messaging.

5.1 Speech Engines & API

Initially, we started out using CMU's Sphinx engine, as it is the state-of-the-art in HMM but due to limitations in training and time constraints, we switched to another popular engine provided by Microsoft.

5.2 Speech API Overview

The SAPI application programming interface (API) dramatically reduces the code overhead required for an application to use speech recognition and text-to-speech, making speech technology more accessible and robust for a wide range of applications.

5.2.1 API for Text-to-Speech

Applications can control text-to-speech (TTS) using the ISpVoice Component Object Model (COM) interface. Once an application has created an ISpVoice object (see Text-to-Speech Tutorial), the application only needs to call ISpVoice::Speak to generate speech output from some text data. In addition, the IspVoice interface also provides several methods for changing voice and synthesis properties such as speaking rate ISpVoice::SetRate, output volume ISpVoice::SetVolume and changing the current speaking voice ISpVoice::SetVoice

Special SAPI controls can also be inserted along with the input text to change real-time synthesis properties like voice, pitch, word emphasis, speaking rate and volume. This synthesis markup sapi.xsd, using standard XML format, is a simple but powerful way to customize the TTS speech, independent of the specific engine or voice currently in use.

The IspVoice::Speak method can operate either synchronously (return only when completely finished speaking) or asynchronously (return immediately and speak as a background process). When speaking asynchronously (SPF_ASYNC), real-time status information such as speaking state and current text location can polled using ISpVoice::GetStatus. Also while speaking asynchronously, new text can be spoken by either immediately interrupting the current output (SPF_PURGEBEFORESPEAK), or by automatically appending the new text to the end of the current output.

In addition to the ISpVoice interface, SAPI also provides many utility COM interfaces for the more advanced TTS applications.

5.2.2 API for Speech Recognition

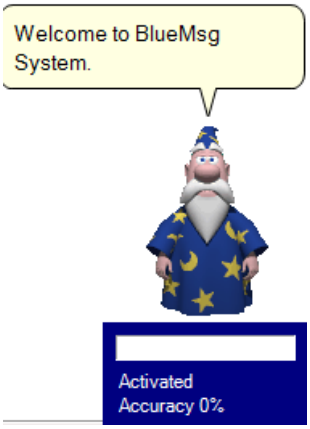
Just as ISpVoice is the main interface for speech synthesis, ISpRecoContext is the main interface for speech recognition. Like the ISpVoice, it is an ISpEventSource, which means that it is the speech application's vehicle for receiving notifications for the requested speech recognition events.

An application has the choice of two different types of speech recognition engines (ISpRecognizer). A shared recognizer that could possibly be shared with other speech recognition applications is recommended for most speech applications. To create an ISpRecoContext for a shared ISpRecognizer, an application need only call COM's CoCreateInstance on the component CLSID_SpSharedRecoContext. In this case, SAPI will set up the audio input stream, setting it to SAPI's default audio input stream. For large server applications that would run alone on a system, and for which performance is key, an InProc speech recognition engine is more appropriate. In order to create an ISpRecoContext for an InProc ISpRecognizer, the application must first call CoCreateInstance on the component CLSID_SpInprocRecoInstance to create its own InProc ISpRecognizer. Then the application must make a call to ISpRecognizer::SetInput (see also ISpObjectToken) in order to set up the audio input. Finally, the application can call ISpRecognizer::CreateRecoContext to obtain an ISpRecoContext.

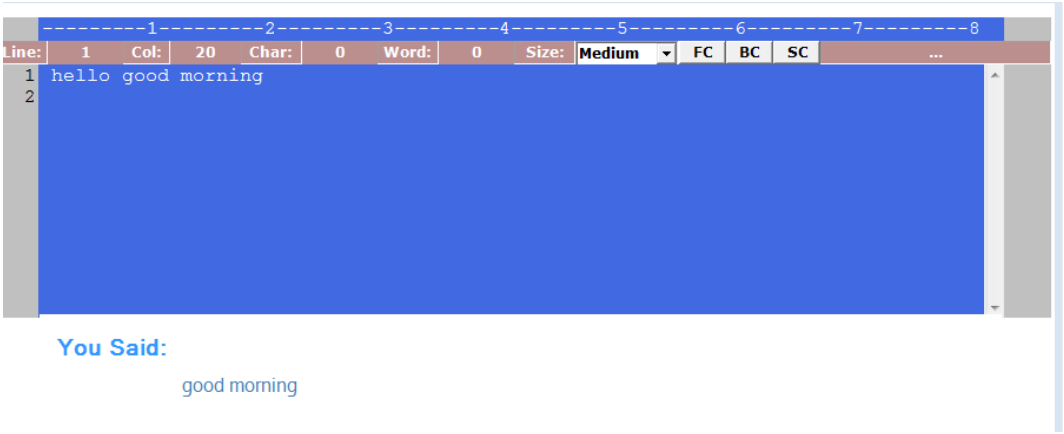
5.2.3 Graphical User Interface

GUI has been designed to accommodate three simple steps for hands-free texting.

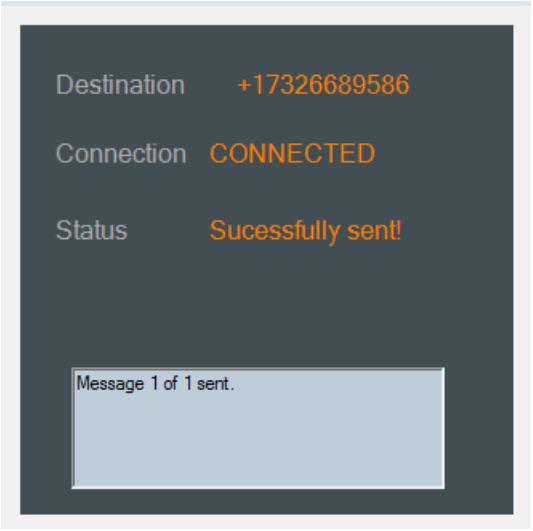
- say "activate" This is will populate a text pad which will take the dictation of the text



- dictate text & say "done" - Upon hearing "done", program will populate a contacts cards



- say the name or phone number & say "send now" - if the name is in the database then it will pull up the phone number. Upon hearing "send now", SMS module will activate and send out the message through wireless modem.



6 Results

Speech processing is not a robust technology and indeed challenging. Results vary drastically dueto the presence of noise in the input data, variation in voice data due to speaker's physical condition, mood etc. Identifying those boundary conditions(as discussed earlier) becomes very hard.
Intuitively, a system with limited vocabulary(grammar) will perform robustly and accurately but in an application such as ours, limited vocabulary is inefficient. Table show the accuracy rate for different sets of grammar rules (in a quite environment) :

Vocabulary	**Accuracy**
15	94%
40	87%
150	77%
2000	59%

Above data was collected in a relatively quite environment. In a noisy environment(presence of more voices) distorts the above results and accuracy rate goes down. We experienced similar limitation during our demo. Due to the weak signal strength inside the WINLAB, we had to present our system outside and the accuracy rate was not as good as it should have been.
Another limitation for this system is provided by the wireless signal strength. If the signal strength is low, SMS transmission will fail. It works fine given relatively full signal strength.

7 Cost Analysis

7.1 BlueMsg Cost

This system is mainly driven through algorithms and softwares. Necessary hardwares for communication and data processing are either built-in or readily available.

7.1.1 Development & Production Cost

- License for commercial usage of Speech Algorithms $\$500 - \$1000$

We can market this as a free product but attach the Google Ads to the software where every unique usage of the software will be paid to us by google .

7.2 Custom processor for car

Six major chips used in the system total of $25.

- $8 application chip from Freelance semiconductor
  - It offers microcontroller solutions, such as the components of embedded control systems, including embedded processors, microcontrollers, embedded microprocessors, and digital signal processors that are used in automotive, consumer, industrial, and computer peripheral applications.
- $5 microcontroller chip from Freescale semiconductor
- $4.80 worth memory chips from Micron Technology
- $3.80 flash memory chip from Samsung
- $1.65 audio chip from Cirrus Logic

With the cost of software licensing and hardware and assuming the markup of three to four times cost typical to the automotive electronics business, the estimated total to manufacture this system will be **$100 - $150**

8 Various other platforms and adaptations

This system can not only support SMS but emails and a variety of different applications. It can be adapted to send out wireless emails using the PC, control music on ipods, and to allow bluetooth enabled devices to be voice-activated.

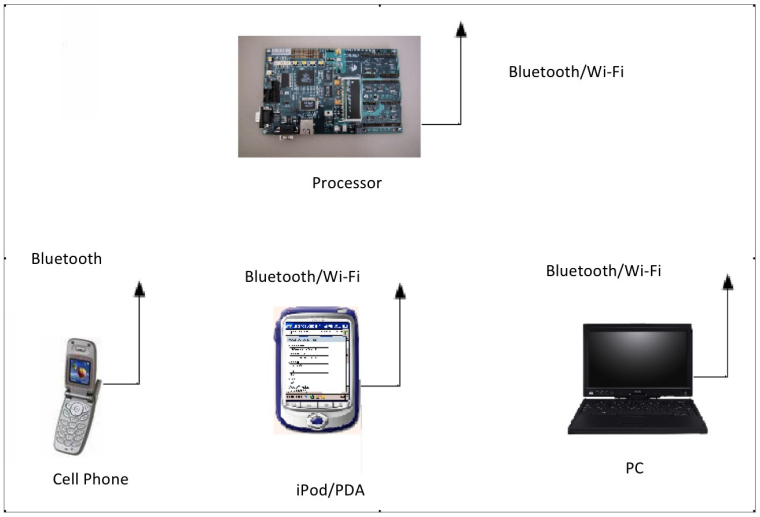


Figure 6: Various Adaptations

References

[1] Wikipedia: The Free Encyclopedia. 22 Jul. 2004. Bluetooth. 10 Aug. 2004 <http://en.wikipedia.org>

[2] Rabiner, Lawrence. *A Tutorial on Hidden Markov and Selected Application in Speech Recognition.* IEEE, 1989.

[3] iSupply: Applied Market Intelligence 2008 <http://www.isupply.com>

[4] Jonathan Allen, M. Sharon Hunnicutt, Dennis Klatt, From *Text to Speech: The MITalk system.* Cambridge University Press: 1987. ISBN 0521306418

[5] Silverman, K.; Beckman, M.; Pierrehumbert, J.; Os tendorf, M.; Wightman, C.; Price, P.; Hirschberg, J., 1992. ToBI: A standard scheme for labeling prosody. *International Conference on Spoken Language Processing.* Canada: Banff, 867-879.

[6] Willie Walker, Paul Lamer. *Sphinx-4: A Flexible Open Source Framework for Speech Recognition.* SUN Microsystem, 2004

[7] Yarrington, D., Pennington, C., Bunnell, T., Gray, J., Lilley, J., Nagao, K., & Polikoff, J. B. (2008). ModelTalker Voice Recorder (MTVR) - A System for Capturing Individual Voices for Synthetic Speech. *Paper presented at the ISAAC 13th Biennial Conference*, Montreal, Canada.