Department of Electrical and Computer Engineering

Voice-Interface to an On-Line Dictionary

In partial fulfillment of the requirements for

EE 4012 Senior Design

By

Mary E. Weber
weber@isip.msstate.edu

Instructor:

Dr. Nail

Spring Semester 1996

Table of Contents

Table of Figures

VOICE INTERFACE TO AN ON-LINE DICTIONARY

Mary Elizabeth Weber

EE 4012 Senior Design Project
Department of Electrical and Computer Engineering
Mississippi State University
Mississippi State, Mississippi 39762
weber@isip.msstate.edu

ABSTRACT

In an era of natural language recognition machines, access to electronic equipment through a speech interface will go a long way towards making state-of-the-art technology available to a larger class of users. A typical application that would be useful to a significant group of people (e.g. students) is an on-line dictionary that can be accessed and queried using voice commands. Currently, no such dictionaries exist for UNIX-based computer systems. Although some personal computers offer this feature to a limited extent, these are constrained by the amount of memory required for a large vocabulary recognition system. In this project, we design an interface that uses public-domain speech-recognition software to recognize specified words and access a dictionary that is available on-line. The resulting system will be publicly available through the ISIP home page.

1. INTRODUCTION

This paper describes the design of an on-line dictionary interface, giving the break down and performance of each part of the system. By making it easier to find a word's definition, this project should help to make writing easier. When designing a voice interface query, it is necessary to determine each section and how it works before integrating it into a system. The parts needed in the database query include the recorded and formatted signal, the speech recognizer, and the dictionary. Publicly available speech recognition software and on-line dictionary were chosen for this design. These parts of the system were not designed as modular functions, which produced a challenge when integrating the pieces. Intermediate steps needed to be developed to take the output of one system and fit it to the input of another.

The most important part of the voice-interface to a dictionary is the speech recognizer. What the recognition portion does is create a test bed for other database queries such as looking up a book

in a library, a phone number in a telephone directory, or a particular movie in a television listing. Designing a database query with a natural language interface is more convenient for the user than employing a complicated programming language such as is currently being used for queries. A natural language interface can be designed to perform a more complicated queries. For example, finding all the books pertaining to the 100 year war and the high ranking officers that survived that war. Designing a program for undertaking such a query would be a complicated task even for an experienced programmer.

1.1. Choosing a Speech Recognizer

However, with an adept speech recognizer such a task can be made possible with far less work and help make the system available to a broader group of users. This voice interface can be designed for availability to speaking into a microphone at your computer or by dialing your phone to indicate a the query. Performing such complicated queries requires access to a reliable recognition system. The number of publicly available natural recognition systems is limited.

The one chosen for this project was the Abbot Hybrid Connectionist-HMM Large-Vocabulary Recognition System. When choosing this system, it was necessary to understand how a state-of-the-art recognition system performs.

The job of a speech recognition system is to understand and be able to respond correctly to a spoken phrase and not only transcribing what is being said by the speaker. To accomplish such a task, presents a number of challenges for the system including word spacing, coarticulation (the preceding or succeeding sounds in a string), the context, the dialect, the speaking rate, and the speaking style. All of these factors affect the performance of the recognizer when one is concerned with making an affordable and small system that runs in real time. At this point, speech recognition is limited by these challenges and the adaptability of a system to different environments. The first step in solving the recognition problem is gaining an understanding of the complexities involved. Research of the human speech process is only just beginning to understand the relaying of neural signal from the inner ear to the higher auditory centers in the brain. When designing a recognizer system one needs to answer the following questions:

1. Does the system need to recognize more than

one speaker?

2. What is the necessary size of the working vocabulary?
3. Should the speech be entered as words with discrete pauses or continuous utterances?
4. How much ambiguity is available in the vocabulary (e.g. “affect” and “effect”)?
5. In what types of environment will the system be operated (e.g. noisy or controlled)?
6. What linguistic constraints and knowledge are included in the system? Where the constraints are concerned with how the fundamental units are put together (the order or the context).

When looking over performances of recognition systems on a yearly basis, experience shows how the recognition process has improved even while the complexity of the systems increases with improvements in computer system speeds and memory. Figure 1 represents the progress accomplished over nearly a decade. It shows how the number of words the recognizer misses decreases with time even as the word lists increase from 1,000 to 20,000 words. Even as the level of recorded speech becomes more natural for the speaker, the recognition systems are still improving.

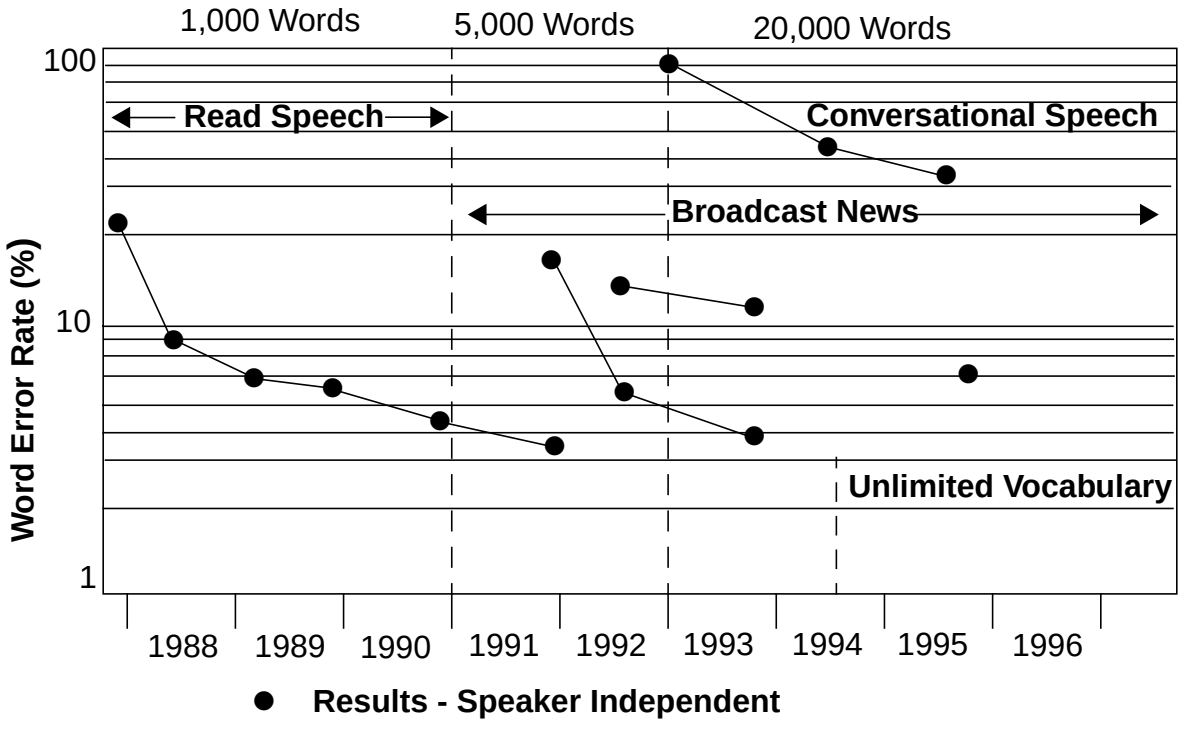

Figure 1: Progress in Speech Recognition

Taking into consideration the complexity of speech recognition systems, the Abbot recognition system was chosen for its availability without a licensing fee. The next portion of this paper explains how the Abbot recognition system performs.

2. THE ABBOT HYBRID CONNECTIONIST-HMM LARGE-VOCABULARY RECOGNITION SYSTEM

The Abbot is a large-vocabulary system developed at Cambridge University, which uses a recurrent network to estimate the probabilities of different phone classes. It also uses hidden Markov model chains to model the lexical and other constraints in our natural language system. Some of the features of the system include a connectionist model, merging specific presentation of the acoustic context and multiple pronunciations. An advantage of this system is a good performance rate using a context- and gender-independent acoustic model and fewer parameters for the hidden Markov model. The following sections include a break-down of the acoustic and language modeling portions of the system.

2.1. Acoustic Modeling

The recurrent network is used to find a phone recognition by mapping one sequence onto another. This is a useful network because it is able to emulate any state machine. Mapping from a sequence of frames of speech to a sequence of phones, produces a correlation effect between the speech frame and the associated phone label. The outputs of the network are used to estimate the observation probabilities within the language model. The maximum probable phone or word is then extracted using decoding techniques.

Sub-word models and a pronunciation model are necessary to be able to represent every word properly in a large vocabulary. The sub-word units can be represented by phones, triphones, or a syllable based approach. A phone is an acoustic category that corresponds with a phoneme (the smallest unit with a distinguished representation). Phones are also dependent on context variables such as speaking rate. By using a string of phones to represent a word’s pronunciation, the job is reduced to finding the estimated phone probability strings and searching a list of phone strings for the most probable word string. The recurrent network is used in this system to estimate the phone class probabilities which will be incorporated into a Markov model word recognition system. The output of the recurrent neural network is considered the posterior probability of a phone class with the

acoustic information. Given the phone class, Bayes' rule can convert the posterior probability into values to be used by the HMM instead of the typically calculated Gaussian values.

The acoustic model shown in Figure 2 performs

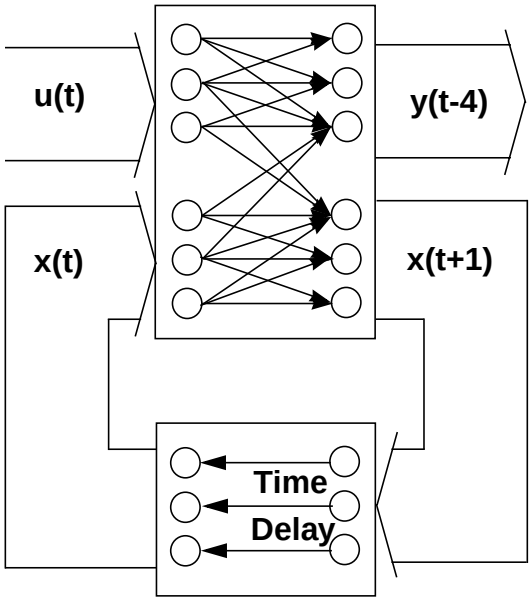

Figure 2: Recurrent Neural Network

so that for each 16 msec frame, an acoustic vector (u(t)) and the current state (x(t)) are presented as inputs to the system. The two vectors are then passed through a standard single layer, feed-forward network. The output vector (y(t-4)) gives an estimation of the posterior probability of each of the phone classes. To take into account the forward acoustic context, the output vector is delayed by four frames (64 msecs). The next state vector shows what is needed for modeling context and the dynamics of the acoustic signal. The output vector is represented by the following equation, where $q_i(t)$ is state i at time t and the input is from time 1 to time t, $u_1^{\,t} = \{u(1)\ldots u(t)\}$.

$$y_i(t) \cong P_r\left(q_i(t) \middle| u_1^{\,t+4}\right)$$

Only a single recurrent network is used for the entire system so that it generates all the phone probabilities in parallel.

2.2. Language Model

The function of the language model is to use the word matcher (HMM) when taking the data from the front end of the recognizer and trying to make words by using phonetic rules, vocabulary, and syntax rules, which are stored in the system. The purpose of the syntax rules is to specify the sequences of phonemes and words allowed so that it provides a model of what is being recognized. A language is made up of strings of phonemes which are formed from a grammar. A grammar is made up of a vocabulary, a set of syntactic types, and a set of generating rules.

These elements serve to take the utterance from a designated starting point to the final string in a probable manner. A grammar generates the

structure, but not the meaning of the utterance. The hybrid approach uses the hidden Markov model (HMM) to form the time-varying nature of the speech signal. AHMM is used to model a speech utterance where all the symbols have their own probability of being one of the possible output states. With each state there is a probability distribution of all the output symbols. The HMM shows the statistical makeup of a word's observation strings. This particular language model is a Markov process on the words, the words are a Markov process on the phones, and the phones are a Markov process on the states. Language processing deals with recognition of a large pattern by breaking it down into smaller subpatterns. The purpose of the language model is to find a small sub-set so the sentence can be broken down into a signal by using the linguistic processing rules.

The Abbot system, when recognizing a spoken utterance, uses a bigram model and a pronunciation dictionary. In the pronunciation dictionary, the phone set used contains 79 symbols, which takes into consideration three levels of stress and a context- and gender-independent model. The connectionist portion of the system is trained to classify phones. The outputs of the phone classifier are treated as the observation terms of the HMM. Therefore, the system is limited to one state per phone model. A duration model for each phone is used so that the output probabilities are tied across all the states of the phone. The transition topography and probabilities give the phones a duration distribution. A bigram model is used where it is assumed that a dependency exists between two words. Operationally, the system looks at a spoken utterance being broken down into <silence> <two words> <two words>... <two words> <silence>. The bigram language model describes the way to search for the most probable word sequence.

A diagram of how the bigram model works is shown in Figure 3.

2.3. The Abbot Demo

With a better understanding of how the Abbot speech recognizer performs, it is time to down-load the publicly available Abbot Demo. The Abbot Demo is designed as a demonstration of the Abbot system as described earlier. This system is developed by researchers from Cambridge University and its purpose is to understand clearly spoken British and American English in a quiet environment. The system has an initial vocabulary

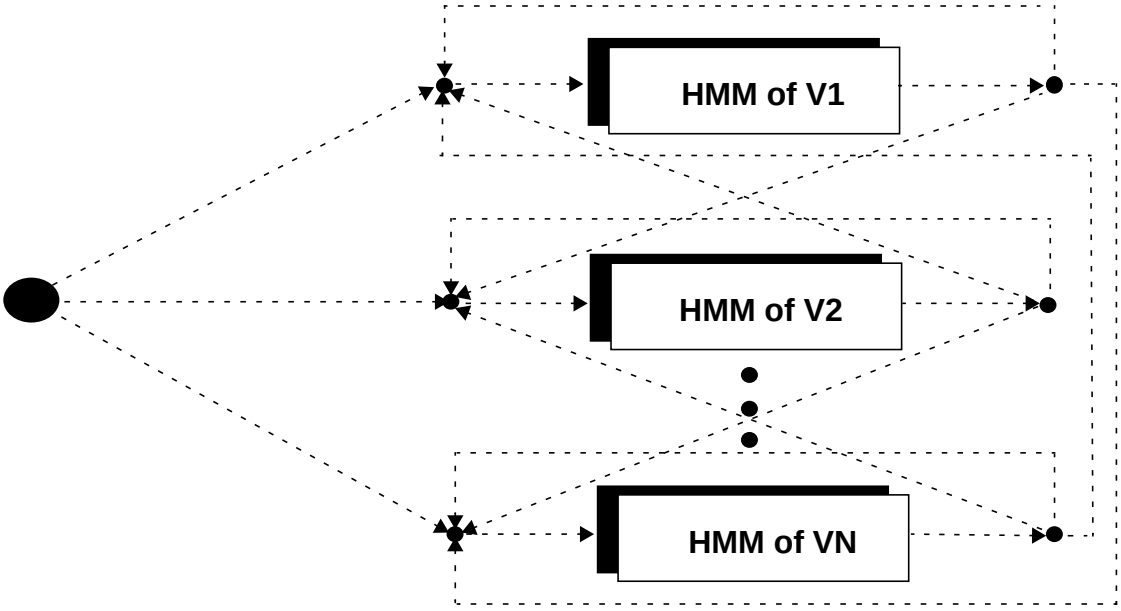

Figure 3: Hidden Markov Model for a Bigram Language Model

of 5,000 words which can be upgraded to 10,000 or 20,000 but this upgrade increases the operation time of the recognizer. If the word spoken is not recognized from among the available 5,000 word vocabulary, it gives an approximation using a string of words. An example of this occurs when the system tries to recognize the word, “experiment”. The response was “X. parent”. In order to make the system FTP accessible, a number of compromises have to be made. The demo uses one recurrent network and context independent models instead of four networks and a context dependent system, which would double the size of the system. This is why the basic language model is limited to 5,000 words. The result of making the system smaller was to make it better for FTP access and disk usage. Increases in the average number of words that were considered as the next word increases the word error rate.

After down-loading the Abbot Demo locally, the only code accessible is the shell script which runs the system. This limited the Abbot Demo to perform like a black box, where the input and output must coincide with what was being used by the system.

With this in mind, the current system can be described.

3. THE SYSTEM DESIGN

In this section each portion of the proposed system

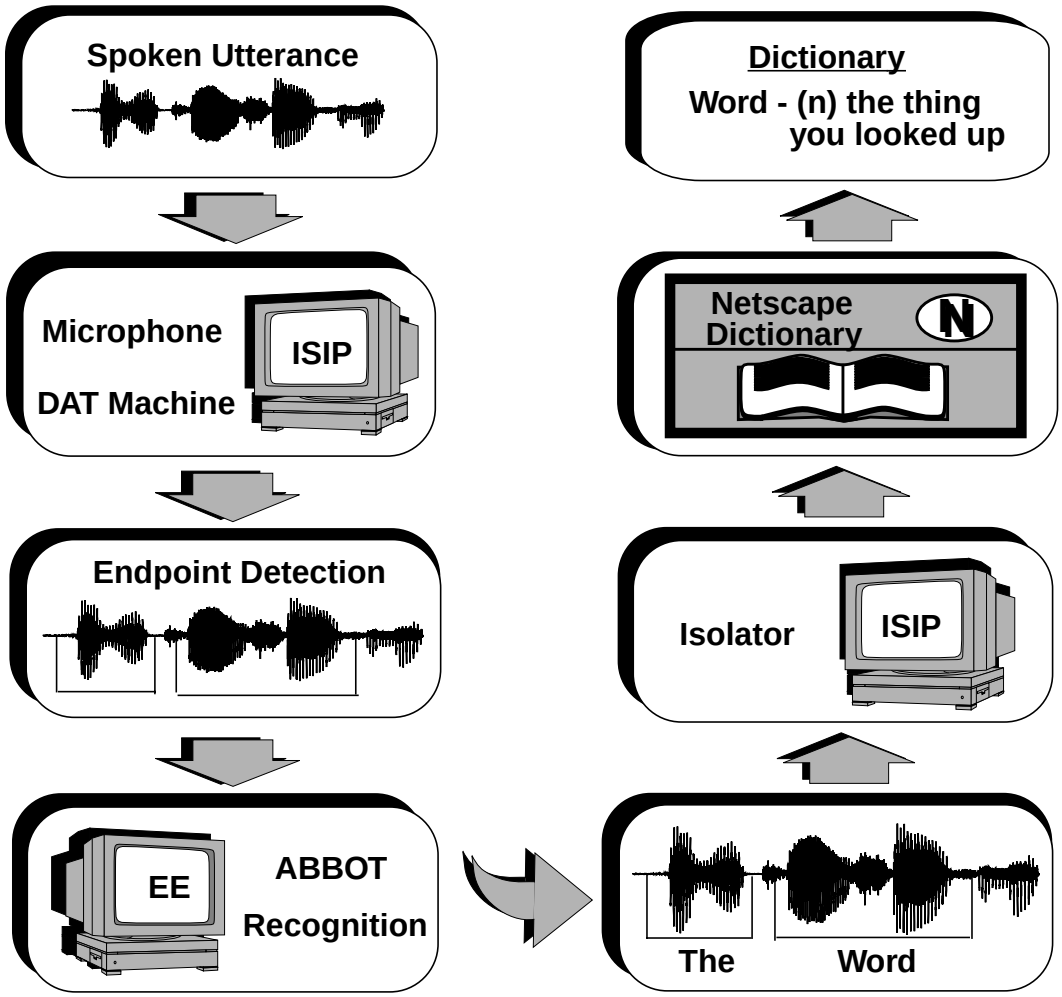

Figure 4: Architecture of the Current System

will be broken down. Figure 4 shows a pictorial view of how the system works.

3.1. Input

The Abbot Demo was capable of taking its input from a file or a microphone. For this project, the input was read from a file because the Abbot Demo will work only on the Sun OS operating system and the microphone system was available on the Solaris operating system. To input a file into the Abbot Demo requires a signal recorded at 16 kHz, ASCII formatted, linearly encoded, and normalize gain. The data recorded from the Digital Audio Tape must be converted from its ".raw" format to ASCII characters through a special program, which also normalizes the gain on the signal. The default recording frequency must be 16 kHz. The signal must also be ready to have its endpoints detected

and excised to get rid of any extraneous noise from the recording.

3.2. Endpoint Detection

The purpose of the endpoint detector is to determine the speech signal when there is background noise during the signal recording. It is also used to determine the beginning and ending of a word. In one approach, the acoustic signal is modeled as a silence (which includes background noise), then the utterance which is subsequently followed by silence. This could cause problems in words that begin with low-energy phonemes making it difficult to distinguish between background noise and the utterance, especially if the background noise is high. Speakers have a tendency to let the energy drop as they speak a word or sigh at the end of a spoken utterance. All of these problems make it difficult to design a generic endpoint detector. When determining the parameters for the endpoint detector, the information available about the noise and the type of utterance must be taken into consideration.

The endpoint detector used for the proposed system was developed during the Fall 1995 Digital Signal Processing course. The endpoint detection algorithms use the energy and the rate of zero crossing for the signal. The digitized speech is first passed through a low-pass filter and the short term energy of the signal is determined. The outputs of the energy calculations are converted into a state machine that depending on the energy levels found. The output from the state machine is then used to determine the endpoint output. The system finds the beginning of an utterance through a stretch of signal states that occurs longer than a set time limit. The end of the utterance is determined when the noise state occurs for a longer period of time than its set time limit. The zero crossing is calculated when the energy of the signal is calculated. Then information is used to help more accurately find the energy signal endpoint. This result helps when determining a word that begins or ends with a low-energy phoneme. When these phonemes are uttered there is a significant amount of zero crossing activity occurs. Once the endpoints are detected on the speech signal, the endpointed speech is passed through an excise signal program. The purpose of this program is to cut off excess background noise from the signal. At this point the signal is ready to be sent to the recognizer on the Sun OS operating system from the Solaris operating system using a perl programming script.

Figure 5 shows a diagram on how the endpoint detections system operates.

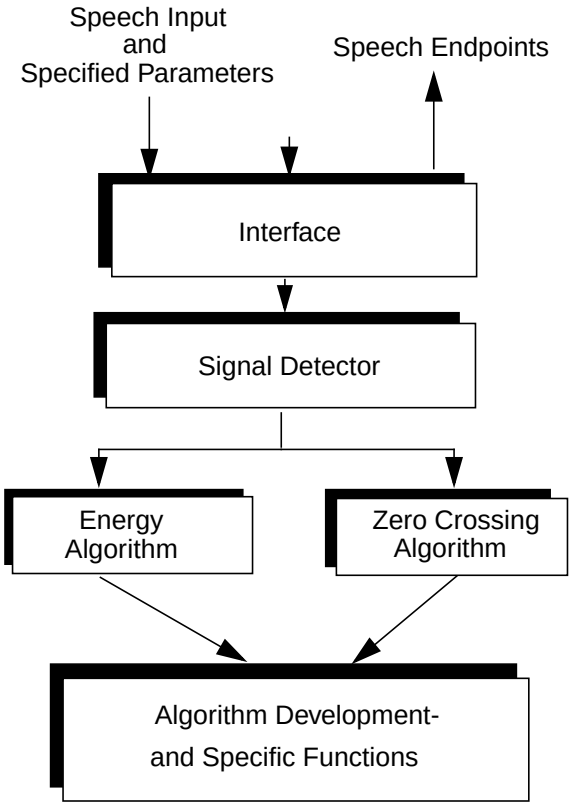

Figure 5: The Endpoint Detection System

3.3. Recognition System

The Abbot Demo inputs the signal manually into the system. From there, the user clicks on the “Pipe to NOWAY” button and starts the recognition process. As the recognition system identifies the speech appropriate signal it prints out to a file the best guess of the word string up to that point of recognition. The final word string is then printed at the end. An example of this occurs when “President Clinton denied it” is spoken. In such a case, the recognizer would output the following:

1

1 THE

1 THE BEST OF TWO

1 THE REST OF THE UNIT

1 PRESIDENT CLINTON DENIED IT

1 PRESIDENT CLINTON DENIED IT A

1 PRESIDENT CLINTON DENIED IT

The recognizer also prints out the amount of time it takes to recognize the speech signal on the last line with the recognized word. The recognition process runs in approximately real time on the Sun OS operating system.

After the system recognizes the utterance, the file with the recognition process is then sent back to the Solaris operating system.

3.4. Isolator

The isolator portion reads the recognition file until it gets to the last line. From here, it reads backward from the beginning of the time segment which was the last thing written to the file, until it gets to the beginning of the line. The repetition number is then stripped from the portion and what was left is the

word originally spoken.

Now the word is ready to be sent along the network to Carnegie Mellon University where the Webster Dictionary interface is located.

3.5. Dictionary

A dictionary is made up of the lexicon, grammar, semantic, phonology, and etymologies of any given word. The lexicon is how a word or words form a larger unit of communication. The grammar associated with a word is how the word is put together. The semantic gives the meaning of the word. The phonology breaks down the word into its phonetics and the etymology is the history of the word. This dictionary has approximately 200,000 words in it. This is a much larger vocabulary than the average 10,000 - 20,000 words used by a working person.

The dictionary can be interfaced through Netscape. It is set up to perform via a point-and-click interface, where the word is typed into the given space, the user hits return, and the dictionary goes to retrieve the required definition. The current interface version being used is the first attempt to make the dictionary available publicly. By purchasing a license, the system can be used locally. The system is also available on CD-ROM, but it can only be accessed through the point-and-click interface rather than through a program interface.

Since the system needed to access the dictionary through a program rather than by the point-and-click-interface, it was necessary to create a shell script that would call Netscape from the command line. This was done using programming commands provided by Netscape.

4. OBSTACLES ENCOUNTERED

After building the system and putting the pieces together, several obstacles related to the performance of the system were encountered. Having the system perform in real time was significant in the overall performance of the system. The two real time constraints affecting the system included not having the recognizer and the dictionary available locally. Sending the data across the network caused a lag in the time performance of the system.

Another major obstacle affecting the operation of the system was the fact that the Abbot recognizer could not recognize every word available in the dictionary. This is to be expected since the

language model is limited to 5,000 words. Also the recognizer is trying to recognize a continuous sequence of two words rather than a single word.

To account for both of these concern, it is necessary to change the language model in the Abbot system to account for a smaller vocabulary size and the ability to recognize every word in the dictionary. A triphone language model would be the ideal choice, because it is made up of a list of three phone sequences that can make up an utterance. Rather than having a word list, the list would contain a list of the three phone sequence possibilities. The list of triphones is considerably shorter than the 200,000 words in the dictionary.

Also, the dictionary can only recognize the root of a word and not the prefix, suffix, or word tense. To solve this problem, a more complicated program will need to be developed to strip the prefix or suffix from the word before sending it to the dictionary. The definition will be displayed including a piece to account for the prefix or the suffix. To get the root from a particular tense one would need to have the dictionary be able to recognize word tenses. Hopefully this will be implemented in the next version of the dictionary.

5. SUMMARY

By breaking down a process and explaining each function, this paper describes the design of the voice-interface to an on-line dictionary. In doing so, this project has brings together a spoken utterance, speech recognizer and dictionary by using intermediate programmed steps.

The future enhancements to the interface software will include accessing the recognizer and dictionary locally. A local recognizer is currently being built for the Solaris operating system during the graduate level speech recognition course. Versions of the dictionary being obtained more conveniently should be available.

6. ACKNOWLEDGEMENTS

First of all, I would like to thank Dr. Joseph Picone for all of time he took helping me to understand how this project is supposed to work. He pushed me hard enough to do something worthwhile rather than just another project. Basically, he gave me a good kick in the pants when I needed to get moving. I would also like to thank the following people for helping me along the way with my programming skills and their much appreciated support on this project: Neeraj Deshmukh, Rick Duncan, Arvind Ganapathiraju, Sean Lauderdale,

and Daniel Williams (my fellow counterparts at ISIP). Lastly, I would like to thank Tony Robinson of Cambridge University for willingly answering my questions pertaining to the Abbot recognizer, which helped point me in the right direction.

REFERENCES

1. A.J. Robinson, *An Application of Recurrent Nets to Phone Probability Estimation*, in *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 298-305, March 1994.
2. C. L. Wayne, *Continuing Technical Challenges: Hubs 3, 4, 5, and 6,* in the transactions of the *ARPA Speech Recognition Workshop,* National Security Agency, Ft. Meade, Maryland, USA, February 1996.
3. D.B. Roe and J.G. Wilpon editors, *Voice Communication Between Humans and Machines*, National Academy Press, Washington D.C., USA, 1994.
4. D. J. Kershaw, A. J. Robinson, and S. J. Renals, The 1995 Abbot Hybrid Connectionist-HMM Large-Vocabulary Recognition System, in the transactions of the *ARPA Speech Recognition Workshop*, Cambridge University, Cambridge, England, February 1996.
5. D. O'Shaughnessy, *Speech Communication: Human and Machine*, Addison-Wesley Publishing Co., Reading Massachusetts, USA, 1987.
6. J.G. Proakis and D.G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, *2nd Edition*, Macmillan, New York, New York, USA, 1992.
7. J.R. Deller, J.G. Proakis, and J.H.L. Hansen, *Discrete Time Processing of Speech Signals*, MacMillan, New York, New York, USA, 1993.
8. K. F. Lee and H. W. Hon, *Speaker-Independent Phone Recognition Using Hidden Markov Models*, in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 11, pp. 1641-1648, November 1989.
9. L.R. Bahl, F. Jelinek, and R.L. Mercer, *A Maximum Likelihood Approach to Continuous Speech Recognition,* in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, no. 2, pp. 179-190, March 1983.
10. L. Rabiner and B.H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1993.
11. L.R. Rabiner and R.W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1978.
12. *Merriam Webster's Collegiate Dictionary 10th Edition,* Merriam-Webster, Inc., Massachusetts, USA, 1993.
13. M. M. Hochberg, S. J. Renals, and A. J. Robinson, *Abbot: The Cued Hybrid Connectionist-HMM Large-Vocabulary Recognition System,* in the *Proceedings of the Spoken Language Technology Workshop*, Cambridge University, Cambridge, England, March 1994.
14. M. M. Hochberg, S. J. Renals, and A. J. Robinson, *Large-Vocabulary Continuous Recognition System Using a Hybrid Connectionist HMM System,* in the *Proceedings of the International Conference on Spoken Language Processing,* Cambridge University, Cambridge, England, 1994.
15. P. C. Woodland, M. J. F. Gales, D. Pye and V. Valtchev, *The HTK Large-Vocabulary Recognition System for the 1995 ARPA H3 Task*, in the transactions of the *ARPA Speech Recognition Workshop*, Cambridge University, Cambridge, England, February 1996.
16. S. Furui, *Speaker-Independent Isolated Word Recognition Using Dynamic Features of Speech Spectrum*, in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-34, no. 1, pp. 52-59, February 1986.
17. S. J. Renals and M. M. Hochberg, Efficient *Search Using Posterior Phone Probabilities Estimates, in the Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Cambridge, England, 1995.
18. T. W. Parsons, *Voice and Speech Processing*, McGraw-Hill Book Company, New York, USA, 1987.
19. V.V. Digalakis, M. Ostendorf, and J.R. Rohlicek, *Fast Algorithms for Phone Classification and Recognition Using Segment-Based Models,* in *IEEE Transactions on Signal Processing,* vol. 40, no. 12, pp. 2885-2896, December 1992.