Programmable Alarm Clock

EEL6586 Project

Student: John Barrios
Instructor: Dr. John Harris
Date: 28 Apr 03

Abstract:

Speech recognition concepts have been implemented with many different applications over the years to help simplify the way we work, play, and live. Applications include things like security, automatic typing, telephone menu navigation, etc. The attempt of each system is to provide an easy, reliable method of performing a task using human speech for the user interface. Another example of the use of speech recognition systems could be tied to a simple alarm clock. In this instance, a user could issue speech commands to the alarm clock in order to program the alarm, the time or perform other possible functions. In this project, the goal will be to simulate the performance of programming a simple alarm clock with speech inputs. The idea is to provide a system that is speaker independent and can be programmed by anyone who can speak the English language commands. This concept could also be extended to a number of other applications. Each application could be used to help simplify and improve the way we perform various tasks.

Project Description:

In this project, I would like to associate the speech recognition techniques learned in class with a practical application. An alarm clock that can be programmed vocally is an example of such a real-world application. In this example, existing speech recognition techniques can be implemented to provide the user with a convenient method for setting the alarm, the time, and even turning the alarm on and off through vocal commands. A limited vocabulary set is needed for the proposed system and isolated word recognition can be assumed to avoid complexity and minimize computational costs. For this type of application, speaker independence is essential, and the system should be designed such that high recognition accuracy can be achieved for any English-speaking user. In this project, I will utilize existing speech recognition techniques (i.e. MFCC/HMM) and will develop a set of algorithms for implementing this limited vocabulary, speaker independent speech recognition system. The algorithms for performing the speech recognition will be designed and tested in Matlab.

System Description:

The system proposed for this project is an alarm clock that can be vocally programmed by the user. The proposed system will operate with the notion that the user will simply be required to depress a switch which activates the recording system. The proposed system would have an embedded microphone, and the speech input would be digitized by an analog to digital converter and stored in memory for future feature extraction and word classification. For ease of use, the system will be designed such that the user will be able to issue all commands in one utterance, however it is expected that the words must be clearly pronounced and have adequate temporal spacing in order to adequately perform end-point detection to isolate each individual word of the utterance. The user of the proposed system will be required to issue a function command utterance (i.e. “Alarm”, “Time”, “Snooze”) followed by one, two or three time commands (i.e. “Twelve”, “Thirty”, “Two”) and an “AM” or “PM” designator for the Time and Alarm commands. A typical example utterance might be, “Time… Twelve… Fourteen… AM.” For this example, the speech recognition system would interpret what was commanded, and the time would be set to 12:14 AM. Another example might be, “Snooze… Fifteen.” In this particular example, the user could command the alarm clock to snooze for fifteen extra minutes.

Once the record button is released and the command has been issued, the speech recognition system will go to work. The process will begin with an end-point detection algorithm that will extract each individual word from the utterance. Once the words are isolated, the Mel-Frequency Cepstral Coefficient (MFCC) features of the each word will be extracted and each word will be classified using a Hidden Markov Model (HMM). The HMM will be trained by a relatively small database of the required system words, and both male and female training words will be used to attempt to make the system speaker independent. Upon performing the speech recognition of each word in the utterance, the resulting command will be programmed into the alarm clock and the setup is complete.

With the above system description in mind, for this project, the operation of the proposed system will be simulated in Matlab. In this case, the previously recorded test utterances will be run in the system and the output will display the function command, the time command/s, and the “AM”/”PM” designator at the end of the simulation. In this way, the alarm clock settings can be simulated and the results of the speech recognition system can be compared to the known test utterance for accuracy.

In order to implement the proposed system, a total of 29 words will need to be able to be recognized. A list of the words needed in this system is shown in Table 1 below.

Command Type	Word
Function	Alarm, Time, Snooze
Time	Zero, One, Two, Three, Four, Five, Six, Seven, Eight, Nine, Ten, Eleven, Twelve, Thirteen, Fourteen, Fifteen, Sixteen, Seventeen, Eighteen, Nineteen, Twenty, Thirty, Forty, Fifty
Designator	AM, PM

Table 1: Required Words for Alarm Clock Programming

For each command utterance, it is possible that a total of two, three, four or five words will need to be extracted from the utterance and speech recognition will need to be performed on each word. A slight pause will be required in between each word to simplify the end-point detection algorithm and promote word isolation. The goal of the project is to have the output of the simulation displayed in the Matlab command window with the function, time, and designator matching the words input through the test utterance.

In order to be realistic and keep the cost and complexity low on the proposed system, the system will be expected to operate with low cost components and realistic system limitations. Therefore, assuming low speed A/D's, a low sampling rate will be assumed. In this case, an 8 kHz sampling rate will be used by the system for recording the speech commands. Also, an 8-bit quantizer will be used for the proposed system implementation. Although these constraints may not be ideal for speech recognition accuracy and performance, this will help to eliminate processing and memory costs in the proposed alarm clock system and would thus keep the product lower cost and more marketable for the average consumer.

Simulation:

In order to implement the proposed alarm clock system as a simulation in Matlab, several different functions had to be performed. The first step was to generate a database for training the system. For this database, a total of four individuals were used for all 29 possible words (See Table 1 for the list of words). Each word was uttered two times by each individual, and the total training set consisted of a total of 232 utterances (This is a relatively small training set, and more training data would be required in an actual system.)

The next step was to generate the code in Matlab to simulate the operation of the proposed alarm clock system. The first function (train.m) was used to create training data for classification of each word of the system. This function extracts the MFCC features [4] from each utterance from the training database and creates a set of training data using Cappe's HMM classifier [5]. The training set was loaded into a database file traindata.mat which is used later for word classification. A second function (wordextract.m) was used to isolate each word in the test utterance and break up the words in order to perform the task of isolated word recognition. This code completes the process of isolating each word and prepares each word for the word recognition which is performed in the last function. The end point detection is performed using the total number zero crossings in 10ms windows to indicate the start and end points for each word in the utterance. In addition to this rough word isolation, further end point detection was experimented with by using the code obtained from the Matlab Auditory Demonstrations (MAD) created by the Speech and Hearing Group at the University of Sheffield, UK (http://www.dcs.shef.ac.uk/~martin/MAD/docs/mad.htm) [6]. This code, however, was not utilized in the final implementation of the simulation due to unknown implementation issues. The third and final function (test.m) was used for isolated word recognition of each of the words isolated from the word extraction function. The purpose of this code is to take each word from the test utterance, classify it, and display the results of the speech recognition system.

Results:

Overall, the results of the classifier were good. Through experimentation, high recognition accuracies were achieved when testing training words and test words that were manually processed to obtain the proper start and end points of each word. However, when testing on words that were extracted by the wordextract.m function, the end-point detection algorithm, the recognition performance was not always as reliable. In the three test utterances provided for testing (testutter1, testutter2, and testutter3), all words were correctly identified by the classifier. For instance in the case of testutter1.wav, the following test commands were given in the utterance:

“Alarm, Six, Zero, Five, AM”

The output of the test.m code dispayed the following results in the command window.

Here are the recognized commands for the Alarm Clock system:

Command 1 is: 'Alarm'

Command 2 is: 'Six'

Command 3 is: 'Zero'

Command 4 is: 'Five'

Command 5 is: 'AM'

In this instance, the wordextract.m function was able to isolate each word from the utterance, and the test.m function was able to correctly classify each word. However, the end point detection algorithm did not always perfectly extract the individual commands from the utterance. Certain sounds, like the ‘oo’ in “Snooze”, often confused the algorithm to select the untrue end points for the word. The resulting errors also provoked errors in classifying the individual words. Further refinement of the word extraction/end point detection algorithms would be extremely beneficial for the final implementation of the proposed alarm clock system.

In addition, the classifier was not always completely accurate when classifying the individual words even with proper end point designation. Although the simulation performed extremely well when using test words that were also used in creating the training database, the algorithm did have slightly reduced classification accuracy for test words not used in the training database. A shortcoming of this experiment was due to the low number of subjects and utterances used to create the training database. The process of preparing words for the database limited the size of the database that was feasible for the purposes of this project. Larger databases of more various utterances would be needed to implement this system for use among the larger population. Additionally, the final testing of the algorithms was accomplished with test utterances that were delivered by an individual that also was used in the training data. To truly test the accuracy of the system, independent test utterances would be needed to test the range of the systems capabilities. Another shortcoming is that in the limited scope of this project, no tests were performed to determine the accuracies involved with this system. Such studies would be required for real-world implementation of this system to provide a capability that would allow this type of speech recognition system to be valuable for the end user. Poor speech recognition performance of this type of system could be the death of its value to the average consumer. Therefore, studies to better understand the expected level of performance would be extremely beneficial.

Conclusion:

In this project, the goal was to produce an example implementation of a real-world system that utilizes speech recognition techniques learned during the semester. The proposed alarm clock system was designed such that the user could easily issue commands to the system by simply depressing a recording button and speaking a set of commands into a recording device. From there, the goal was to simulate the operation of the proposed alarm clock system by extracting individual words from the utterance and then classifying each word using a previously generated training set. Despite some problems, the system was able to operate as designed, and the result demonstrated potential to make this into a realizable system. However, further refinement of the algorithms designed in this project would be necessary to make the system robust and reliable enough to handle the environment and user diversity of a system that could be marketed on a large scale. Some possible improvements and areas for further research and evaluation include: better word isolation/end point detection techniques, use of different feature sets and classifiers, larger databases of utterances for training, pre-processing/filtering of the speech inputs, etc. With more time and research, the proposed system could be further refined and potentially implemented in a real system. However, despite some of the limitations of this project, the goal to experiment with the possible functionality of an alarm clock that can be programmed using speech commands from the user was successfully met. Overall, the system was able to perform speech recognition of various speech inputs simulating the operation of the proposed alarm clock system. More importantly, I was able to learn a great deal about the various approaches used to help solve the problems faced in the design of speech recognition systems and the challenges confronted in their ultimate real-world implementation.

References:

[1] J.A. du Preez, Approaches to speaker independent isolated word recognition, Southern African Conference on Communications and Signal Processing, pp. 41-45, 24 Jun 1988

[2] L. Lamel, L Rabiner, et al, An improved endpoint detector for isolated word recognition
IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol 29, Issue 4 , pp. 777-785, August 1981

[3] L. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE, Vol. 77 No. 2, pp 257-286, February 1989

[4] Malcolm Slaney, Auditory Toolbox, http://rvl4.ecn.purdue.edu/~malcolm/interval/1998-010/, 1998

[5] Oliver Cappe, H2M Matlab Toolkit, http://www.tsi.enst.fr/~cappe/h2m/h2m.html, 1995

[6] Martin Cooke and Guy Brown, Matlab Auditory Demonstrations, http://www.dcs.shef.ac.uk/~martin/MAD/docs/mad.htm, 1999