AUTOMATIC SPEECH RECOGNITION SYSTEM FOR CONTROLLING A ROBOTIC SYSTEM USING ROMANIAN

Alexandru Goloca[1], Pentiuc Stefan-Gheorghe[2]

The "Stefan cel Mare" University 13th, University Street, Romania,
tel.: +40-230-216147, E-mail: [1]alexg@eed.usv.ro, [2]pentiuc@eed.usv.ro

Abstract

In this paper, we describe a proposed Automatic Speech Recognition system that can be used to give vocal commands to a robotic system. The system was specially created to use Romanian as the main language with respect to its particularities: specific phonetic rules and large variety of accents.

The main purpose of the system is to act as a touch-less interface for a demonstrative robotic system. This interface was designed to allow a remote human user to give simple commands to the robotic system without any physical contact with the robot.

There are more ways to create an Automatic Speech Recognition System but, given the complexity of the task, the methods that were used had to be reliable and this is the reason while the Hidden Markov Models approach was chosen.

The current status of the project allows a remote user to give simple commands to the experimental robot using a microphone, a laptop or a PDA and wireless connection.

Introduction

Computer Speech recognition, also known as Automatic Speech recognition (ASR) enables a computer, or a computer guided device to recognize spoken words in an automated way, with no human aid. ASR offers the possibility to create new touch-less interfaces for computers and this further opens the door for new developments in many areas of interest. One particular area is the medical field, where speech recognition techniques can be implemented on high-end surgical equipment and another is robotics, where a human can control a robot using nothing but the voice.

The current status

All Speech recognition consists of two problems. They are not completely separated but have different degree of difficulty, as presented in [2] and [6].

The first and "simple" problem is the isolated word recognition. The problem is not really that simple. It assumes that there is an utterance which is known to contain a single word. The requirement is to identify the one word contained in the spoken data. The second and most complex problem is contextual speech recognition. This involves recognizing words inside a proposition, propositions inside a phrase and even phrases. This "complex problem" is not discussed in this paper but it should be mentioned that this approach is used by complex commercial ASRs.

The Accuracy of the system is one key factor and it is measured in word error rate.

Another classification is made from the degree of independence that the system possesses, or how much do the accuracy of the system depends on the user:

- User dependent speech recognition systems: they offer good results for a particular speaker (the one that was used for training the system) but less good results if another person is using the system. If the other person has very different voice parameters, the results are poor.
- User independent speech recognition systems: they offer good results for any category of speakers (from a global point of view) but not very good results for particular categories of speakers.

There are three main speech recognition techniques (as presented in [1]) that have been developed during the past years and some of them are used to develop hybrid models. These main techniques are:

1. Dynamic Time Warping (DTW);
2. Artificial Neural Networks (ANN);
3. The Hidden Markov Model (HMM);
4. Hybrid models.

Dynamic Time Warping

This is, according to [1], the first technique used but it offers poor results in ASR. It is important only from a historical point of view and is not used in nowadays large scale applications.

Artificial Neural Networks

If Artificial Neural Networks are to be used as a tool for Automatic Speech Recognition, the input dataset consists of a series of speech vectors. These vectors contain relevant information about a pronounced word. A training set is presented to the system and the system is adjusting itself in such a way that would generate a correct classification of the input data (the word that was pronounced is already known for the training set). After

that was done, the system is able to receive test data (the words that need classification) and identify the word that was pronounced.

Hidden Markov Model

The approach based on Hidden Markov Models represents the main choice for speech recognition applications in our days (according to [1], [2] and [3]) and it seems that future developments are still possible. HMMs are used in many types of applications where there is a need to classify objects. These applications can vary from speech recognition to handwriting recognition, posture recognition and so on.

A HMM represents a statistic model in which the modeled system is assumed to be a "Markov Process" with unknown parameters. The main challenge is to compute the hidden parameters knowing only the visible parameters (those that can be observed). The extracted parameters can be used after that for several different purposes, but the main goal is to build a system that is able to classify objects. From the ASR point of view spoken words are objects that have to be associated with a class and Hidden Markov Models are able to perform that.

A typical HMM was illustrated in Fig.1. The meaning of its parameters is:

- x - Hidden states: these cannot be directly observed but their effect exists and can be observed directly;
- y - System outputs: these can be directly observed and represent the effect of internal state transitions;
- a - Transition probabilities: these represent the probabilities for the system to pass from state to state;
- b - Output probabilities: these represent the probabilities to obtain certain values for system outputs.

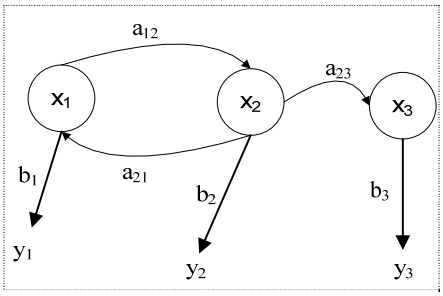

Fig.1. A HMM and the meaning of its parameters

There are three main problems (challenges) involving HMMs:

1. The model parameters are given and the requirement is to compute the probability to obtain a certain sequence as system output. This problem is solved using an algorithm called "*The forward-backward procedure*", an advanced algorithm which comes from the "dynamic programming" family of algorithms.
2. The model parameters are given and the requirement is to compute the most plausible sequence of states that generated the observed output sequence. This problem is solved using the Viterbi algorithm (presented in [7] and [8]). This algorithm allows retracing the most plausible sequence of states that has been used in the training stage. This is the "*recognition problem*" or the classification.
3. One output sequence is given, or even a set of sequences and the requirement is to compute the most likely internal transitions and output probabilities that allowed for the output sequence to occur. This has the meaning of "*training the model*". This problem is solved by the Baum-Welch algorithm (as described in [9]). This algorithm adjusts the requested probabilities until the output sequence coincides with the desired one.

Hybrid Models

These are the "top models" and they offer probably the best results, being used in commercial applications. They are based on both ANN and HMM, according to [1] and [2].

An Artificial Neural Network is used to identify parts of a word (called phonemes). After the phonemes have been identified, they feed multiple HMMs. These HMMs which will compute the most probable word made from those phonemes. More detailed: several HMMs are used: there is one for every word that could possibly be recognized. All models offer a probability (the probability that the observed word was generated by the current model) and the maximum value from all those models will indicate the most likely word (the model that really produces that word is expected to offer a very high value for that probability).

A proposed system based on HMM

This proposed system uses the following steps in order to accomplish classification of the spoken words:

- Sound acquisition**:** this step is made of more sub-stages, like: converting the sound waves into electric analogue signal (done with a microphone), followed by the conversion of analogue signal to discrete

numeric values using an ADC (Analogue to Digital Converter) circuit. At this step is very important that the surrounding environment has a low noise level;

- Preprocessing raw data: also consists of sub-stages:
  1. Signal normalization and segmentation: the level of noise is detected using different methods, and considered to represent "silence signal". The signal then is sliced in small portions called *windows* or *frames*. These may contain data representing from 20 to 50 milliseconds from the digital signal.
  2. Digital filtering makes sure that all spikes appearing at the beginning and the end of each frame will not affect the final result. In order to obtain that effect, a multiplication with a Hamming window is used. After the Hamming filtering cut the unwanted spikes from the beginning and the end of each frame, it's time for another filter: a low-pass filter.
  3. Word boundaries detection: At this step words are being separated from silence in order to be extracted for analysis.
- Feature extraction: at this very important stage one must choose *what is meaningful* from the preprocessed signal. This is needed because using the entire signal from a frame is not going to provide enough useful information and hence an extraction method is needed. There are some relevant parameters that could be used but the most popular approaches are:
  1. Linear Prediction Coding (LPC): uses some polynomial approximations but is not a very popular method;
  2. Cepstral Processing: a very popular method which uses Mel Fourier Cepstral Coefficients (MFCC), described in [10] and was used in [11]. Obtaining these coefficients means using many complicated operations like Discrete Fourier Transform (as presented in [5]), Logarithmical operations and Discrete Cosine Transform. The required steps are presented in **Ошибка! Источник ссылки не найден.Ошибка! Источник ссылки не найден.**:

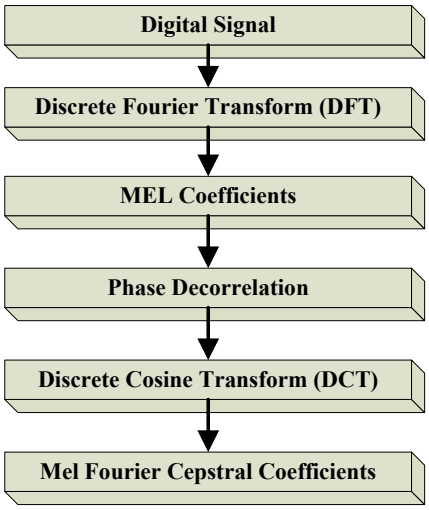

Fig.2. Obtaining the MFCCs

- Training the HMM is achieved using the Baum-Welch algorithm. Inner parameters of the HMM are adjusted until it will generate the desired output sequence.
  A codebook can be used at this step and it ensures a very fast operating when it will come to actually recognizing the word (see next step). In order to build the codebook Vector Quantization can be performed on the speech vectors extracted at the Feature Extraction step. A codebook contains a number of points from a n-size space and a number of centroids (weight centers, also n-sized points);
- Classification using HMMs. This is done using the Viterbi algorithm.

The current project. Proposed extensions

The goal of the current project is to create a system capable of offering a similarity measure for a spoken word. More precise, a person is supposed to pronounce a certain command word and the system is supposed to evaluate the pronunciation and decide what the given command was. Further, the command is sent to a robot for execution (e.g. change of direction, acceleration and so on).

Most of the methods already presented are being used in this project. The chosen software platform was Java, due to its high portability (can work on PCs as well as on embedded systems that offer Java support) and very rich libraries (known as *packages*).

The implemented training stages (up until now) are:

- Sound acquisition;
- Windowing is the action of slicing the signal in equal sized frames. This is done using overlapping windows.
- Digital filtering: a Hamming window and a Low-pass filter are used for each window in order to ensure a clean signal for the following steps.
- Word boundary detection is done using the zero-crossing rate with threshold method since it offers good results and is able to deal with both low-frequency and medium-frequency noise.
- Feature extraction is performed using Cepstral Processing. After a complex succession of steps, 12 Mel Fourier Cepstral Coefficients are extracted from a frame of 512 elements.
- A codebook is built for every trained word using Vector Quantization (VQ) applied on the MFCCs already obtained. In order to build the codebook two algorithms had to be used K-means and Binary Split;
- Codebooks can be stored on files after they have been built and they can be loaded whenever they are needed (training has to be done just once for a training set, not each time recognition is needed).

The implemented recognition steps are as follows (in italics those that are the same as for training):

- *Sound acquisition;*
- *Windowing;*
- *Digital filtering;*
- *Word boundary detection;*
- *Feature extraction;*

Distance computing and recognition. At this final step a vector containing relevant speech data (the Mel Fourier Cepstral Coefficients) is confronted with all existing codebooks, previously loaded from files.

A distortion is calculated for each codebook. This is the sum of Euclidian Distances between the points belonging to the current word and those belonging to the centroids in the codebook. After computing all the distortions, the minimum distortion is found and the codebook that featured that distortion is assumed to indicate the most likely word.

Using other kinds of distances, experiments have been performed, in order to find whether they would provide a better measure of similarity. Some distances used were: Manhattan Distance, Cebisev Distance or Mahalanobis Distance.

Conclusions

There are some differences between the classical speech recognition problem and the current project. The most important ones could be:

The need for an accurate recognition: Unlike dictation software that would simply write text, this system is supposed to control the actions of a robot using commands. This gives it great responsibilities since its actions might affect people (e.g. the robot could hit people while moving as a result of deficient understanding of commands).

There is the need to expand the number of possible commands in the future when the complexity of the robot might increase. This should be done without rewriting the speech recognition software.

There are a number of difficulties that arise from the fact that the main used language is not English but Romanian, as it had been shown in [12]. This means that phonetics and phonetic rules are not the same as English ones and the available English libraries can't be used.

Acknoledgements

This research was financed by the 56-CEEX (TERAPERS) and 131-CEEX (INTEROB) research grants.

References:

[1] "Speech Recognition", http://en.wikipedia.org/wiki/Speech_ recognition
[2] Young Steve, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying (Andrew) Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, Phil Woodland, The HTK Book. COPYRIGHT 2001-2006 Cambridge University Engineering Department.
[3] "Hidden Markov Model", http://en.wikipedia.org/wiki/ Hidden_Markov_model
[4] Smith, Steven W., "The Scientist and Engineer's Guide to Digital Signal Processing", Second Edition, California Technical Publishing, San Diego, California
[5] "Discrete Fourier Transform", http://en.wikipedia.org/wiki/Discrete_Fourier_ transform
[6] Rabiner L. R., "A tutorial on hidden Markov models and selected applications in speech recognition"
[7] Viterbi Andrew J., "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", IEEE Transactions on Information Theory 13(2):260–269, April 1967. (Section IV.)
[8] "Viterbi Algorithm", http://en.wikipedia.org/wiki/Viterbi_ algorithm
[9] "Baum-Welch Algorithm", http://en.wikipedia.org/ wiki/Baum_Welch_algorithm
[10] "Cepstrum", http://en.wikipedia.org/wiki/Cepstrum,
[11] Iwanashi, Naoto: Active and Unsupervised Learning for Spoken Word Acquisition Through a Multimodal Interface
[12] Chivu C. 2007. Applications of Speech Recognition for Romanian Language. Advances in Electrical and Computer Engineering, Suceava, Romania 1/2007, volume 7 (14), pp. 29-3