MINI PROJECT REPORT

ON

“SPEECH RECOGNIZER”

Submitted in

Partial Fulfillment of requirements for the Award of Degree

*of*

Bachelor of Technology

*In*

Computer Science and Engineering

By

**(Group Number: 8)**

# Aditya Singh (1816410025)

# Akash Rajput (1816410033)

# Anay Singh (1816410050)

# Abhinav Mishra (1816410010)

Under the supervision of

**Rajat Verma**

**(Assistant Professor)**



**Pranveer Singh Institute of Technology**.

 Kanpur - Agra - Delhi National Highway - 2

Bhauti - Kanpur - 209305.

(Dr. A.P.J. Abdul Kalam Technical University)

**Objective**

The prime objective of the project being proposed is to design and build a system that a basic user can interact so that she/he can make use of voice commands to deal with system i.e. making a system that has capability of recognizing the isolated speaker words and process the request to forward the given task

The typical objectives are listed below:

• To make use of domain specific models and algorithms in field of speech recognition.

• To develop an interactive voice response system along with speech recognition attribute.

• To understand the basics of speech processing.

• To get knowledge on various speech recognition approaches.

• To get insights on speech responsive application development.

Academically, the project is primarily focused on fulfilling the discipline of an engineering student as a computer engineer working on a project and gain experience as a team throughout the different phases of a project. Some Typical academic objectives of the project are:

• To fulfill the requirements of the major project of B.E. in computer engineering.

• To design and complete a functional project that integrates various course concepts.

• To develop various skills related to project management like team work, resource management, documentation and time management.

• To get hands-on experience of working in a project as a team work.

• To learn about and become familiar with the professional engineering practices.

**Introduction**

Most of the works done till today on the field of IVR system has been primarily focused on the input mechanisms based on the keyboard or touch pad. In such cases it is tedious to provide the input command every time through typing of texts. This way of providing input to the computer system may be enhanced if we could provide direct speech input instead of typing. This enables in fast interaction between the system and user and therefore increases overall satisfaction of the customers. This also increases the speed of access of the information from the system. Furthermore, English language has been widely implemented in IVR systems. This has created difficulty for people while interacting with the system. Thus by implementing the Nepali voice commands it is easier to interact and provide the input to the system.

The major focus of the project being developed is the use of direct Nepali voice command for the interactive voice response system without need of typing which then further can be applicable to real world applications like call centers, customer support systems and other several organization inquiry systems.

**Literature Review**

**Historical Overview of Speech Recognition**

The concept of speech recognition started somewhere in 1940s, practically the first speech  
recognition program appeared in 1952 at the bell labs, that was about recognition of a digit  
in a noise free environment. 1940s and 1950s consider as the foundational period of the  
speech recognition technology, in this period work was done on the foundational paradigms  
of the speech recognition that is automation and information theoretic models. In the 1960s,  
it was able to recognize small vocabularies (order of 10-100words) of isolated words, based  
on simple acoustic-phonetic properties of speech sounds. The key technologies that were  
developed during this decade were, filter banks and time normalization methods. In 1970s  
the medium vocabularies (order of 100-1000 words) using simple template-based, pattern  
recognition methods were recognized. In 1980s large vocabularies (1000-unlimited) were  
used and speech recognition problems based on statistical, with a large range of networks  
for handling language structures were addressed. The key invention of this era were model  
(HMM) and the stochastic language model, which together enabled powerful new methods  
for handling continuous speech recognition problem efficiently and with high performance.  
In 1990s the key technologies developed during this period were the methods for stochastic  
language understanding, statistical learning of acoustic and language models, and the methods  
for implementation of large vocabulary speech understanding systems. After the five decades  
of research, the speech recognition technology has finally entered marketplace, benefiting  
the users in variety of ways. The challenge of designing machine that truly functions like an  
intelligent human is still a major one going forward.

**Speech Recognition Overview**

Speech Recognition (SR) is the process of extracting the string of words automatically from  
the speech signal, by means of an algorithm. It is the ability of a machine or program  
to identify words and phrases in spoken language and convert them to a machine readable  
format. Speech recognition is a powerful tool of the information exchange using the acoustic  
signal. Therefore, not surprisingly, the speech signal is for several centuries the subject of  
research. Speech recognition is a technology that makes a computer able to capture the words  
spoken by a human with a help of microphone. These words are later on recognized by  
speech recognizer, and in the end, system outputs the recognized words which can also serve  
as input to the further systems to accomplish several task. Speech recognition is basically the  
6  
science of talking with the computer, and having it correctly recognized. Speech recognition  
is getting the meaning of an utterance such that one can respond properly whether or not  
one has correctly recognized all of the words. ASR has always been considered as an  
important bridge in fostering better human to human and human to machine communication.  
In the past, however, speech never actually became an important modality in the human to  
machine communication. This is partly because the technology at that time was not good  
enough to pass the usable bar for most real world users under most real usage conditions, and  
partly because in many situations alternative communication modalities such as keyboard  
and mouse significantly outperform speech in the communication efficiency, restriction, and  
accuracy. In the recent years, speech technology started to change the way we live and work  
and became one of the primary means for humans to interact with some devices. This trend  
started due to the progress made in several key areas. First, Moor’s law continues to function.  
The computational power available today, through multi-core processors, general purpose  
graphical processing units (GPUs), and CPU/GPU clusters, is several orders of magnitude  
more than that available just a decade ago. This makes training of more powerful yet complex  
models possible. These more computation demanding models significantly reduced the error  
rates of the ASR systems. Second, we can now access to much more data than before,  
thanks to the continued advance of the Internet and the cloud computing. By building  
models on big data collected from the real usage scenarios, we can eliminate many model  
assumptions made before and make systems more robust. Third, mobile devices, wearable  
devices, intelligent living room devices, and in vehicle infotainment systems became popular.  
On these devices and systems, alternative interaction modalities such as keyboard and mouse  
are less convenient than that in the personal computers. Speech, which is the natural way  
of human to human communication and a skill that majority of people already have, thus  
becomes a more favorable interaction modality on these devices and systems.

**Human Ear and Speech Recognition**

Humans are more effective than machines at recognizing speech. This advantage for human  
listeners is particularly pronounced for speech that is heard against background noise, contains  
unfamiliar words or is degraded in other ways. Yet, automatic speech recognition (ASR)  
systems have made substantial advances over the past few decades and are now in everyday  
use by millions of people around the world. Until the performance of automatic speech  
recognition (ASR) surpasses human performance in accuracy and robustness, we stand to  
gain by understanding the basic principles behind human speech recognition (HSR). In this  
section we provide a brief explanation of how human hearing works and how it is modeled.  
We will discuss in brief the functionality of several components and try to understand the  
7  
relation of it with the speech recognition systems.

**Human Hearing system**

The main function of hearing system is to get information about the outside, which is carried  
by pressure variations in the air, that is, sound wave. Sound waves are generated by the  
movement or vibration of an object, that is, sound source. As the vibrating object moves  
out and in, the nearby air molecules create a slight increase and decrease in pressure, called  
condensation and rarefaction, respectively. From the pressure variations, we perceive what  
the sound source is and where it comes from. We perceive a sound wave, which is a continual  
time series signal, by the ears. We also perceive three-dimensional acoustic space by the ears,  
mainly because the head-related transfer function (HRTF) between a point of a sound source  
and the two ear entrances has directional characteristics from the shapes of the head and the  
pinna. The pinna significantly modify the incoming sound, particularly at high frequencies,  
and this is important in our ability for sound localization.After a sound wave arrives nearby,  
it passes through the peripheral auditory system, the outer ear, middle ear, and inner ear.  
Figure 2.1: Human Ear Hearing System  
8  
1. The Outer Ear  
The outer ear is the external part of the auditory system, including the pinna and the  
ear canal. Sound travels down the ear canal and causes the eardrum, or tympanic  
membrane, to vibrate. Because of the resonance of the outer ear, we are more sensitive  
to sound frequencies between 1000 and 6000 Hz. The pinna is the only visible part of  
the ear (the auricle) with its special helical shape. It is the first part of the ear that reacts  
with sound. The function of the pinna is to act as a kind of funnel which assists in  
directing the sound further into the ear. Without this funnel the sound waves would take  
a more direct route into the auditory canal. This would be both difficult and wasteful as  
much of the sound would be lost making it harder to hear and understand the sounds.  
The pinna is essential due to the difference in pressure inside and outside the ear. The  
resistance of the air is higher inside the ear than outside because the air inside the ear is  
compressed and thus under greater pressure. In order for the sound waves to enter the  
ear in the best possible way the resistance must not be too high. This is where the pinna  
helps by overcoming the difference in pressure inside and outside the ear. The pinna  
functions as a kind of intermediate link which makes the transition smoother and less  
brutal allowing more sound to pass into the auditory canal (meatus).Once the sound  
waves have passed the pinna, they move two to three centimetres into the auditory canal  
before hitting the eardrum, also known as the tympanic membrane. The function of the  
ear canal is to transmit sound from the pinna to the eardrum. The eardrum (tympanic  
membrane), is a membrane at the end of the auditory canal and marks the beginning  
of the middle ear. The eardrum is extremely sensitive and pressure from sound waves  
makes the eardrum vibrate. The auditory canal functions as a natural hearing aid which  
automatically amplifies low and less penetrating sounds of the human voice. In this  
way the ear compensates for some of the weaknesses of the human voice, and makes it  
easier to hear and understand ordinary conversation.  
2. The Middle Ear  
The middle ear is the part of the ear between the eardrum and the oval window. The  
middle ear transmits sound from the outer ear to the inner ear. The middle ear consists  
of three bones: the hammer (malleus), the anvil (incus) and the stirrup (stapes), the  
oval window, the round window and the Eustrachian tube.The eardrum is very thin,  
measures approximately eight to ten millimeter in diameter and is stretched by means  
of small muscles. The pressure from sound waves makes the eardrum vibrate. The  
vibrations are transmitted further into the ear via three bones in the middle ear. These  
three bones form a kind of bridge, and the stirrup, which is the last bone that sounds  
reach, is connected to the oval window. The oval window is a membrane covering the  
entrance to the cochlea in the inner ear. When the eardrum vibrates, the sound waves  
9  
travel via the hammer and anvil to the stirrup and then on to the oval window. When  
the sound waves are transmitted from the eardrum to the oval window, the middle ear  
is functioning as an acoustic transformer amplifying the sound waves before they move  
on into the inner ear. The pressure of the sound waves on the oval window is some 20  
times higher than on the eardrum.  
The pressure is increased due to the difference in size between the relatively large  
surface of the eardrum and the smaller surface of the oval window. The round window  
in the middle ear vibrates in opposite phase to vibrations entering the inner ear through  
the oval window. In doing so, it allows fluid in the cochlea to move. The Eustachian  
tube is also found in the middle ear, and connects the ear with the rearmost part of the  
palate. The Eustachian tube’s function is to equalize the air pressure on both sides of  
the eardrum, ensuring that pressure does not build up in the ear. The tube opens when  
you swallow, thus equalizing the air pressure inside and outside the ear.  
3. The Inner Ear  
The inner ear is the innermost part of the ear, which consist of the cochlea, the  
balance mechanism, the vestibular and the auditory nerve. Once the vibrations of the  
eardrum have been transmitted to the oval window, the sound waves continue their  
journey into the inner ear. The inner ear is a maze of tubes and passages, referred to as  
the labyrinth. In the labyrinth can be found the vestibular and the cochlea.  
In the cochlea, sound waves are transformed into electrical impulses which are  
sent on to the brain. The brain then translates the impulses into sounds that we know  
and understand.The cochlea resembles a snail shell or a wound-up hose and is filled  
with a fluid called perilymph and contains two closely positioned membranes. These  
membranes form a type of partition wall in the cochlea. However, in order for the fluid  
to move freely in the cochlea from one side of the partition wall to the other, the wall  
has a little hole in it (the helicotrema). This hole is necessary, in ensuring that the  
vibrations from the oval window are transmitted to all the fluid in the cochlea.  
The auditory nerve is a bundle of nerve fibres that carry information between the  
cochlea in the inner ear and the brain. The function of the auditory nerve is to transmit  
signals from the inner ear to the brain.The hair fibres in the cochlea are all connected to  
the auditory nerve and, depending on the nature of the movements in the cochlear fluid,  
different hair fibres are put into motion. When the hair fibres move they send electrical  
signals to the auditory nerve which is connected to the auditory centre of the brain.  
In the brain the electrical impulses are translated into sounds which we recognise and  
10  
understand. As a consequence, these hair fibres are essential to our hearing ability.  
Should these hair fibres become damaged, then our hearing ability will deteriorate.  
The vestibular is another important part of the inner ear. The vestibular is the  
organ of equilibrium. The vestibular’s function is to register the body’s movements,  
thus ensuring that we can keep our balance. The vestibular consists of three ringshaped passages, oriented in three different planes. All three passages are filled with  
fluid that moves in accordance with the body’s movements. In addition to the fluid,  
these passages also contain thousands of hair fibres which react to the movement of the  
fluid sending little impulses to the brain. The brain then decodes these impulses which  
are used to help the body keep its balance.  
**2.3.2 Sound Signal Transduction Mechanism**In the human ear the basiliar membrane is contained within cochlea that supports thousands  
of sensory cells which forms the cochlear nerve. It is one of the innermost part of the ear.  
The basiliar membrane acts as a frequency spectrum analyzer. When exposed to a high  
frequency signal, the basilar membrane resonates where it is stiff, resulting in the excitation  
of nerve cells close to the oval window. Likewise, low frequency sounds excite nerve cells at  
the far end of the basilar membrane. This makes specific fibers in the cochlear nerve respond  
to specific frequencies. This organization is called the place principle, and is preserved  
throughout the auditory pathway into the brain. Also the principle called volley principle  
is used for the transduction purpose of the sound signal arriving the human ear. Here a  
nerve cell on the basilar membrane can encode audio information by producing an action  
potential in response to each cycle of the vibration. For example, a 200 hertz sound wave can  
be represented by a neuron producing 200 action potentials per second. However, this only  
works at frequencies below about 500 hertz, the maximum rate that neurons can produce  
action potentials. The human ear overcomes this problem by allowing several nerve cells to  
take turns performing this single task. For example, a 3000 hertz tone might be represented  
by ten nerve cells alternately firing at 300 times per second. This extends the range of the  
volley principle to about 4 kHz, above which the place principle is exclusively used.  
Table below shows the relationship between sound intensity and perceived loudness.  
It is common to express sound intensity on a logarithmic scale, called decibel SPL (Sound  
Power Level). On this scale, zero dB SPL is a sound wave power of 10-16w*atts*/*cm*2, about  
the weakest sound detectable by the human ear. Normal speech is at about 60 dB SPL, while  
painful damage to the ear occurs at about 140 dB SPL.  
11  
Figure 2.2: Human Auditary System  
The difference between the loudest and faintest sounds that humans can hear is about  
120 dB, a range of one-million in amplitude. Listeners can detect a change in loudness when  
the signal is altered by about one dB (a 12 per cent change in amplitude). In other words, there  
are only about 120 levels of loudness that can be perceived from the faintest whisper to the  
loudest thunder. The sensitivity of the ear is amazing; when listening to very weak sounds,  
the ear drum vibrates less than the diameter of a single molecule. The perception of loudness  
relates roughly to the sound power to an exponent of 1/3. For example, if you increase the  
sound power by a factor of ten, listeners will report that the loudness has increased by a factor  
of about two (101/3 ≈ 2). This is a major problem for eliminating undesirable environmental  
sounds, for instance, the beefed-up stereo in the next door apartment. Suppose you diligently  
cover 99 per cent of your wall with a perfect soundproof material, missing only one per cent  
of the surface area due to doors, corners, vents, etc. Even though the sound power has been  
reduced to only 1 per cent of its former value, the perceived loudness has only dropped to  
about 0.011/3 ≈ 0.2, or 20 per cent. The range of human hearing is generally considered to  
be 20 Hz to 20 kHz, but it is far more sensitive to sounds between one kHz and four kHz. For  
example, listeners can detect sounds as low as 0 dB SPL at three kHz, but require 40 dB SPL  
at 100 hertz (an amplitude increase of 100). Listeners can tell that two tones are different if  
their frequencies differ by more than about 0.3 per cent at three kHz. This increases to three  
percent at 100 hertz. The primary advantage of having two ears is the ability to identify the  
direction of the sound. Human listeners can detect the difference between two sound sources  
that are placed as little as three degrees apart, about the width of a person at 10 meters. This  
directional information is obtained in two separate ways. First, frequencies above about one  
kHz are strongly shadowed by the head. In other words, the ear nearest the sound receives a  
stronger signal than the ear on the opposite side of the head. The second clue to directionality  
is that the ear on the far side of the head hears the sound slightly later than the near ear,  
12  
Figure 2.3: Audibility range of sound by human ear at different intensity level  
due to its greater distance from the source. Based on a typical head size (about 22 cm)  
and the speed of sound (about 340 meters per second), an angular discrimination of three  
degrees requires a timing precision of about 30 microseconds. Since this timing requires the  
volley principle, this clue to directionality is predominately used for sounds less than about  
one kHz. Both these sources of directional information are greatly aided by the ability to  
turn the head and observe the change in the signals. An interesting sensation occurs when a  
listener is presented with exactly the same sounds to both ears, such as listening to monaural  
sound through headphones. The brain concludes that the sound is coming from the center  
of the listener’s head. While human hearing can determine the direction a sound is from, it  
does poorly in identifying the distance to the sound source. This is because there are few  
clues available in a sound wave that can provide this information. Human hearing weakly  
perceives that high frequency sounds are nearby, while low frequency sounds are distant. This  
is because sound waves dissipate their higher frequencies as they propagate long distances.

**Speech Recognition System**

The idea behind speech recognition is to provide a means to transcribe spoken words into  
written text. There exist many approaches to achieve this goal. The most simple technique is to  
build a model for every word that needs to be recognized.Speech signal primarily conveys the  
words or message being spoken. Area of speech recognition is concerned with determining  
13  
Figure 2.4: Speech Recognition System  
the underlying meaning in the utterance. Success in speech recognition depends on extracting  
and modeling the speech dependent characteristics which can effectively distinguish one word  
from another. The system is a collection of several modules as shown in Figure

**Input Voice Signal**

The preliminary requirement of the IVR system is the input voice signal from the user such that  
the system may interact with the user. Most computer systems have the built in microphone  
facility for this purpose. Also with the help of external microphone the voice signal can  
be input to the system, the PC sound card produces the equivalent digital representation  
of received audio. During the phase of training the speech engine the recorded audio is  
taken and then are used for the sample generation and finally fed into the system model  
for training purpose and for the interactive voice response system the real time input voice  
of the system user is taken using the microphone. The circumstances under which input  
voice signal is uttered plays important role in speech recognition i.e. the factors such as too  
noisy environment, wrong utterance of word etc may diminish the performance of system.  
Therefore the input signal must be as clear as possible for the best results possible.  
14

**Preprocessing stage**

The stage of the speech preprocessing refers to the purification of the input voice signal so  
as to feed it into main speech recognition engine in a suitable format for best outcomes.The  
preprocessing stage in speech recognition systems is used in order to increase the efficiency  
of subsequent feature extraction and classification stages and therefore to improve the  
overall recognition performance. Commonly the preprocessing includes the sampling step, a  
windowing and a de-noising step as shown in Figure below. At the end of the preprocessing  
the compressed and filtered speech frames are forwarded to the feature extraction stage.These  
processes are discussed below in brief.  
Figure 2.5: Input speech preprocessing  
1. Sampling stage  
In order that a computer is able to process the speech signal, it first has to be digitized.  
Therefore the time-continuous speech signal is sampled and quantized. The result is a  
time- and value discrete signal. According to the Nyquist-Shannon sampling theorem  
a time-continuous signal that is band limited to a certain finite frequency fmax needs  
to be sampled with a sampling frequency of at least 2 *fmax*. In this way it can be  
reconstructed by its time-discrete signal. Since human speech has a relatively low  
bandwidth (mostly between 100Hz and 8 KHz) a sampling frequency of 16 KHz is  
sufficient for speech recognition tasks.  
2. Windowing and frame formation  
Speech is a non-stationary time variant signal. We assume that human speech is built  
from a dictionary of phonemes, while for most of the phonemes the properties of  
speech remain invariant for a short period of time ( 5-100ms). In order to obtain  
frames we multiply the speech signal with a windowing function. This windowing  
function weights the signal in the time domain and divides it into a sequence of partial  
signals. By doing so we gain time information of every partial signal keeping in mind  
that an important step of the preprocessing and feature extraction is a spectral analysis  
of each frame.  
15  
3. Denoising stage  
The stage of denoising or noise reduction, also referred to as enhancing of speech  
degraded by noise, aims to improve the speech signals quality. The objective is  
to improve the intelligibility, a measure of how comprehensible speech is. Noise  
corrupting speech signals can be grouped coarsely into the following three classes:  
• Microphone related noise  
• Electrical noise (e.g. electromagnetically induced or radiated noise) and  
• Environmental noise  
The first two types of noise can be easily compensated by training the speech recognizers  
on corresponding noisy speech samples, but compensating the environmental noise is  
not that elementary, due to its high variability.  
Noise is ubiquitous in almost all acoustic environments. The speech signal, that is  
recorded by a microphone is generally infected by noise originating from various  
sources. Such contamination can change the characteristics of the speech signals  
and degrade the speech quality and intelligibility, thereby causing significant harm to  
human-to-machine communication systems.  
Noise detection and reduction for speech applications is often formulated as a digital  
filtering problem, where the clean speech estimation is obtained by passing the noisy  
speech through a linear filter. With such a formulation, the core issue of noise reduction  
becomes how to design an optimal filter that can significantly suppress noise without  
noticeable speech distortion.  
Noise reduction is the crucial step in speech signal processing. Each signal is contained  
with some kind of noise in it which deteriotes the speech signal quality.  
Noise reduction techniques depending on the domain of analyses like Time, Frequency  
or TimeFrequency/Time-Scale.  
The Noise reduction methods are classified into four classes of algorithms: Spectral  
Subtractive, Subspace, Statistical-model based and Wiener-type. Some popular Noise  
reduction algorithms are, The log minimum mean square error logMMSE (Ephraim  
& Malah 1985), The traditional Wiener (Scalart & Filho 1996), The spectral subtraction based on reduced-delay convolution (Gustafsson 2001), The exception of the  
logMMSE-SPU (Cohen & Berdugo 2002), The logMMSE with speech-presence uncertainty (Cohen Berdugo 2002), The multiband spectral-subtractive (Kamath & Loizou  
2002), The generalized subspace approach (Hu &Loizou 2003), The perceptually  
based subspace approach (Jabloun & Champagne 2003), The Wiener filtering based  
on wavelet-thresholded multitaper spectra (Hu & Loizou 2004), Least-Mean-Square  
16  
(LMS), Adaptive noise cancellation (ANC) [3], Normalized(N) LMS, Modified(M)-  
NLMS, Error nonlinearity (EN)-LMS, Normalized data nonlinearity (NDN)-LMS  
adaptation etc. Among those many methods, one of the most simple and effective  
is the spectral subtraction method. It is quite popular method. Spectral Subtraction  
method, subtracts the estimated noise from the original signal to enhance the speech  
recognition. The noise is estimated from the original signal itself and subtracted to  
the original signal, which thus improves the Signal-to-Noise ratio (SNR). It is assumed  
that the signal is distorted by a wide-band, stationary, additive noise, the noise estimate  
is the same during the analysis and the restoration and the phase is the same in the  
original and restored signal.  
Figure 2.6: Noise Removal Process in input voice signal

**Feature Extraction Stage**

After the preprocessing step, feature extraction is the second component of automatic speech  
recognition (ASR) systems. It helps to identify the components of audio signals that are good  
for identifying the linguistic content and discarding all the other stuff such as background  
noise. The speech signal are slowly timed varying signals (quasi-stationary). When examined  
over a sufficiently short period of time, the characteristics of signal remain fairly stationary.  
The information in the speech signal is represented by the short term amplitude of the speech  
17  
signal.The extraction of feature vectors is based on these short term amplitude spectrum of  
speech signals. This component should derive descriptive features from the windowed and  
enhanced speech signal to enable a classification of sounds. The feature extraction is needed  
because the raw speech signal contains information besides the linguistic message and has a  
high dimensionality. Both characteristics of the raw speech signal would be unfeasible for the  
classification of sounds and result in a high word error rate. Therefore, the feature extraction  
algorithm derives a characteristic feature vector with a lower dimensionality, which is used  
for the classification of sounds.  
There are several feature extraction techniques such as Linear Predictive Analysis  
(LPC), Linear Predictive Cepstral Coefficients (LPCC), Mel-Frequency Cepstral Coefficients  
(MFCC) etc. MFCC is the most commonly used feature extraction method in ASR. To  
extract a feature vector containing all information about the linguistic message, MFCC  
mimics the logarithmic perception of loudness and pitch of human auditory system and tries  
to eliminate speaker dependent characteristics by excluding the fundamental frequency and  
their harmonics. Among several features generated we consider only the relevant feature set  
for the classification model. These generated feature set is known as feature vector which  
define mathematical characteristics of a speech signal. Such feature vectors act as a input  
to the classification models such as HMM(Hidden Markov Mode)l, RNN(Recurrent Neural  
Network) etc.

**Mel Frequency Cepstral Coefficients (MFCC)**

**Mel Frequency Cepstral Coefficients** is a popular feature extraction technique in  
speech recognition. The Mel Frequency Cepstral Coefficients are the representation of a  
windowed short term signal derived from the Fast Fourier Transform (FFT) of the signal on  
a non linear mel scale of frequency, which is based on the human ear scale.  
For the computation of MFCC, the speech signal is divided and framed into 20-40ms  
long frames. The frames are overlapped for smooth transitions. The next step is to perform  
Discrete Fourier Transform of the frames. FFT is used to speed up the processing. Then the  
frequencies obtained from the FFT are wrapped onto the mel scale. A mel is a unit of pitch  
defined so that pairs of sounds which are perceptually equidistant in pitch are separated by  
an equal number of mels. The mapping between frequency in Hertz and mel scale is linear  
below 1000 Hz and logarithmic above 1000 Hz. The mel frequency m can be computed from  
frequency as

|  |  |
| --- | --- |
| *mel*„ *f* ” = 1127*ln*„1 + *f* 700 ” | (2.1) |

18  
The mel-spaced filterbanks are computed. This is a set of 20-40(standard is 26)  
triangular filters that we apply to the output of DFT from earlier steps. Then log of the each  
energy in the filterbank is taken. The next step is to calculate Discrete Cosine Transformation  
(DCT) which ranges coefficients according to the significance.

**Linear Predictive Coding(LPC)**

Linear Predictive Coding (LPC) is a powerful speech analysis technique. The basic  
idea behind LPC is that a specific speech sample at the current time can be approximated as  
a linear combination of past speech samples.  
Linear Prediction is the technique of computation of a parametric model based on least  
mean squared error theory. The speech signal is approximated as a linear combination of  
its precious samples. The obtained LPC coefficients describe the formants. The frequency  
at which the resonant peaks occur are called the formant frequencies. Thus, in this method,  
locations of the formants in a speech signal are estimated by computing the linear predictive  
coefficients over a sliding window and finding the peaks in the spectrum of the resulting LP  
filer.

**Perceptual Linear Prediction (PLP)**

The Perceptual Linear Prediction model describes the psychophysics of human hearing  
process more accurately in feature extraction process. PLP, similar to LPC analysis, is based  
on the short-term spectrum of speech. But, PLP modifies the short term spectrum of the  
speech by several psychophysically based transformations to match human auditory system.  
The PLP coefficients are calculated by first carrying out N-point DFT. A frequency  
warping to Bark scale is applied. The critical-band power spectrum is computed through  
discrete convolution of the power spectrum with the piece-wise approximation of the criticalband curve. The smoothed spectrum is down-sampled at intervals around 1 Bark. The three  
steps of frequency warping, smoothing and sampling are integrate into a single filter-bank  
called Bark filter bank. An equal loudness pre-emphasis weight the filter-bank outputs. The  
equalized values are further processed by Linear Prediction (LP). Applying LP to the warped  
line spectrum computes the predictor coefficients of a signal that has this warped spectrum  
as a power spectrum.  
19

**Acoustic Model**

An acoustic model is used in Automatic Speech Recognition to represent the relationship  
between an audio signal and the phonemes or other linguistic units that make up speech. The  
model is learned from a set of audio recordings and their corresponding transcripts. It is  
created by taking audio recordings of speech, and their text transcriptions, and using software  
to create statistical representations of the sounds that make up each word.  
Acoustic model development is process of developing a speech recognition engine to  
recognize speech. The software acoustic model breaks the words into the phonemes. There  
are different popular ways to build this model, some of which are DTW (Dynamic Time  
Warping), HMM (Hidden Markov Model), RNN (Recurrent Neural Networks) etc.

**Hidden Markov Model**

A Markov model is a stochastic model which models temporal or sequential data i.e.  
data that are ordered. It provides a way to model the dependencies of current information  
with previous information. The simplest Markov model is a Markov chain. Markov Chain  
models the state of a system with a random variable that changes through time. In this  
context, the Markov property suggests that the distribution for this variable depends only on  
the distribution of previous state.  
Figure 2.7: A Markov Chain  
A Hidden Markov Model is a Markov chain for which sate is partially observable. The  
observations are typically insufficient enough to precisely determine the state. A Hidden  
Markov Model, is a stochastic model where the states of the model are hidden. Each state  
can emit an output which is observed.  
20  
Figure 2.8: Structure of a Hidden Markov Model

**Recurrent Neural Network**

A recurrent neural network (RNN) is a class of artificial neural network where connections between units form a directed cycle. This allows it to exhibit dynamic temporal  
behavior.  
Figure 2.9: A recurrent neural network and the unfolding in time  
Recurrent networks, on the other hand, take as their input not just the current input  
example they see, but also what they perceived one step back in time. The decision a recurrent  
net reached at time step *t* - 1 affects the decision it will reach one moment later at time step  
*t*. So recurrent networks have two sources of input, the present and the recent past, which  
combine to determine how they respond to new data, much as we do in life. Recurrent  
networks are distinguished from feedforward networks by that feedback loop, ingesting their  
own outputs moment after moment as input. It is often said that recurrent networks have  
memory. Adding memory to neural networks has a purpose: There is information in the  
sequence itself, and recurrent nets use it to perform tasks that feedforward networks can not.  
The purpose of recurrent nets is to accurately classify sequential input. We rely on  
the backpropagation of error and gradient descent to do so. Recurrent networks rely on an  
extension of backpropagation called backpropagation through time, or BPTT. Time, in this  
case, is simply expressed by a well-defined, ordered series of calculations linking one time  
step to the next, which is all backpropagation needs to work.  
21  
Just as a straight line expresses a change in x alongside a change in y, the gradient  
expresses the change in all weights with regard to the change in error. If we can not know  
the gradient, we can not adjust the weights in a direction that will decrease error, and our  
network ceases to learn. Recurrent nets seeking to establish connections between a final  
output and events many time steps before were hobbled, because it is very difficult to know  
how much importance to accord to remote inputs. This is partially because the information  
flowing through neural nets passes through many stages of multiplication.Because the layers  
and time steps of deep neural networks relate to each other through multiplication, derivatives  
are susceptible to vanishing or exploding.

**LSTM RNN :** LSTM stands for Long Short Term Memory. LSTM is a variant of  
RNN which help preserve the error that can be backpropagated through time and layers. By  
maintaining a more constant error, they allow recurrent nets to continue to learn over many  
time steps (over 1000), thereby opening a channel to link causes and effects remotely.  
LSTMs contain information outside the normal flow of the recurrent network in a gated  
cell. Information can be stored in, written to, or read from a cell, much like data in a computer  
's memory. The cell makes decisions about what to store, and when to allow reads, writes  
and erasures, via gates that open and close. Unlike the digital storage on computers, however,  
these gates are analog, implemented with element-wise multiplication by sigmoids, which  
are all in the range of 0 to 1.  
Those gates act on the signals they receive, and similar to the neural networkâĂŹs  
nodes, they block or pass on information based on its strength and import, which they filter  
with their own sets of weights. Those weights, like the weights that modulate input and  
hidden states, are adjusted via the recurrent networks learning process. That is, the cells  
learn when to allow data to enter, leave or be deleted through the iterative process of making  
guesses, backpropagating error, and adjusting weights via gradient descent.  
Figure 2.10: A LSTM Block  
22

**GRU :** GRU stands for Gated Recurrent Unit. A gated recurrent unit (GRU) is  
basically an LSTM without an output gate, which therefore fully writes the contents from its  
memory cell to the larger net at each time step.  
Figure 2.11: A Gated Recurrent Unit Block

**Types of ASR system**

Speech Recognition System are characterized by different parameters. Some of the more  
important of which are discussed here in brief.

**On the basis of input speech signal**

In Speech Recognition System the ability to recognize the speech signal can be subdivided  
into different classes as below:  
1. Isolated Words: In this type, system accepts single utterance at a time. And usually  
requires each utterance to have quiet on both side of sample window and require a  
speaker to wait between words. Its response will be better for single word but give poor  
result for multiple words input.  
2. Connected Words: In this type, multiple words given to the system which runs  
separately as isolated words and having small duration of time between them.  
3. Continuous Speech: In this type, natural speech is spoken by the user that is detectable  
by the machine. Continuous speech recognition is difficult to create because they utilize  
special method for implementation.  
4. Spontaneous Speech: In this type natural and spontaneous word has the ability to handle  
23  
a variety of natural features such as words run together including mispronunciations,  
non-words and false statements, which are difficult to read.

**On the basis of speaker model**

Every speaker has unique properties which affects the voice. On the basis of these properties  
system is divided into two main classes.  
1. Speaker Dependent Model:Speaker dependent model depends on specific speaker.  
These models are easier to implement and less expensive. It gives more accurate result  
for specific speaker and less accurate result for other speakers  
2. Speaker Independent Model:Speaker independent models depend upon many speakers.  
These models are difficult to implement and more expensive. It gives more accurate  
result for many speakers and less accurate result for specific speaker.

**On the basis of type of vocabulary**

1. Small size vocabulary that includes tens of words.  
2. Medium size vocabulary that includes hundreds of words.  
3. Large vocabulary size that includes thousands of words.  
4. Very large size vocabulary that includes tens of thousands of words.  
5. Out of size vocabulary includes mapping a word from the vocabulary into the unknown world

**Feasibility Study**

Speech Recognition is one of the hottest topics in the current field of technology and science. Many researches have been carried out in this field from several decades ago to till today and many of them are still under study to optimize the study. The ASR system have been utilized in several sectors and have proved their importance in today's technological world. By undertaking this project our attempt is to make use of speech recognition in a simple interactive system to automate the task using the voice command. We have undergone through several feasibility studies to make sure that the project is feasible and be developed. Some study topics are discussed below:

**Technical Feasibility**: ASR have been utilized under several platforms and several development approaches have been developed. Development of new artificial intelligence and pattern matching models have made it more, simpler for implementation of ASR embedded with interactive system. Similarly, today's powerful computing processors and easy data collection software makes it more technically feasible.

**Operational Feasibility**: Many researches have been carried out in the field interactive systems using ASR most of which using English language. Such systems have proved to be easily operable several platforms. Our project is just a kind of implementation of ASR to automate task through Nepali voice. Thus, this makes it operationally feasible for development.

**Economic Feasibility**: The project is economically feasible to begin with as no expensive hardware and software components is required. Similarly, all the tools and techniques to be used are open source and are easily available free of cost. Data collection is done among us and other individuals which is economically feasible.

**Schedule Feasibility**: To develop the project a proper time line has been projected to complete relevant portion of the project in scheduled time period. Most of the Necessary resources are searched on the web and are available to begin research in time. Also, all the related software packages are easily available which makes if more feasible.

**Technology Used**

**Python Programming Language**

Python is a general-purpose, open source, computer programming language. It is optimized for software quality, developer productivity, program portability, and component integration. Python is used by at least hundreds of thousands of developers around the world in areas such as internet scripting, systems programming, user interfaces, product customization, numeric programming, and more. It is generally considered to be among the top four or five most widely-used programming languages in the world today.

The major reasons behind the use of python programming language for this project are discussed below in brief.

• **Software quality**: For many, Python's focus on readability, coherence, and software quality in general sets it apart from other tools in the scripting world. Python code is designed to be readable, and hence reusable and maintainable much more so than traditional scripting languages. The uniformity of Python code makes it easy to understand, even if you did not write it. In addition, Python has deep support for more advanced software reuse mechanisms, such as object-oriented programming (OOP) and function programming.

• **Developer productivity**: Python boosts developer productivity many times beyond compiled or statically typed languages such as C, C++, and Java. Python code is typically one-third to one-fifth the size of equivalent C++ or Java code. That means there is less to type, less to debug, and less to maintain after the fact. Python programs also run immediately, without the lengthy compile and link steps required by some other tools, further boosting programmer speed.

• **Program Portability**: Most Python programs run unchanged on all major computer platforms. Porting Python code between Linux and Windows, for example, is usually just a matter of copying a 50 script's code between machines. Moreover, Python offers multiple options for coding portable graphical user interfaces, database/ access programs, web based systems, and more. Even operating system interfaces, including program launches and directory processing are as portable is Python as they can possibly be.

• **Support Libraries**: Python comes with a large collection of pre-built and portable functionality, known as the standard library. This library supports an array of application-level programming tasks, from text pattern matching to network scripting. In addition, Python can be extended with both homegrown libraries and a vast collection of third-party application support software. Python 's third-party domain offers tools for website construction, numeric programming, serial port access, game development, and much. The NumPy extension, for instance, has been described as a free and more powerful equivalent to the MATLAB numeric programming system.

• **Component Integration**: Python scripts can easily communicate with other parts of an application, using a variety of integration mechanisms. Such integrations allow Python to be used as a product customization and extension tool. Today, Python code can invoke C and C++ libraries, can be called from C and C++ programs, can integrate with Java and .NET components, can communicate over frameworks such as COM and Silverlight, can interface with devices over serial ports, and can interact over networks with interfaces like SOAP, XML-RPC, and CORBA. It is not a standalone tool.

**NumPy**

NumPy is the high performance, numeric programming extension for python. It is the core library for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

It contains among other things:

• a powerful N-dimensional array object

• sophisticated (broadcasting) functions

• tools for integrating C/C++ and Fortran code

• useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multidimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

**Pyaudio**

PyAudio provides Python bindings for PortAudio, the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms. PyAudio is inspired by:

• pyPortAudio/fastaudio: Python bindings for PortAudio v18 API.

• tkSnack: cross-platform sound toolkit for Tcl/Tk and Python.

PyAudio provides Python bindings for PortAudio, the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms. PyAudio is inspired by: pyPortAudio/fastaudio: Python bindings for PortAudio v18 API. tkSnack: cross-platform sound toolkit for Tcl/Tk and Python. To use PyAudio, we first instantiate PyAudio using pyaudio.PyAudio() , which sets up the portaudio system. To record or play audio, we open a stream on the desired device with the desired audio parameters using pyaudio.PyAudio.open() . This sets up a pyaudio.Stream to play or record audio. We play audio by writing audio data to the stream using pyaudio.Stream.write(), or read audio data from the stream using pyaudio.Stream.read(). In "blocking mode" each pyaudio.Stream.write() or pyaudio.Stream.read() blocks until all the given/requested frames have been played/recorded. Alternatively, to generate audio data on the fly or immediately process recorded audio data, use the "callback mode".

**Pomegranate**

pomegranate is a python package which implements fast, efficient, and extremely flexible probabilistic models ranging from probability distributions to Bayesian networks to mixtures of hidden Markov models. Pomegranate has been used in our project to create the HMM models.

**PyQT**

PyQt4 is a toolkit for creating GUI applications. It is a blending of Python programming language and the successful Qt library. Qt library is one of the most powerful GUI libraries. PyQt4 is developed by Riverbank Computing.

PyQt4 is implemented as a set of Python modules. It has 440 classes and 6000 functions and methods. It is a multiplatform toolkit which runs on all major operating systems, including UNIX, Windows, and Mac OS. PyQt4 is dual licensed. Developers can choose between a GPL and a commercial license. Previously, GPL version was available only on UNIX. Starting from PyQt version 4, GPL license is available on all supported platforms.

PyQt4’s classes are divided into several modules. Some of them are:

• QtCore

• QtGui

• QtNetwork

• QtXml

• QtSvg

• QtOpenGL

• QtSql

The QtCore module contains the core non-GUI functionality. This module is used for working with time, files and directories, various data types, streams, URLs, mime types, threads or processes. The QtGui module contains the graphical components and related classes. These include for example buttons, windows, status bars, toolbars, sliders, bitmaps, colours, and fonts. The QtNetwork module contains the classes for network programming. These classes facilitate the coding of TCP/IP and UDP clients and servers by making the network programming easier and more portable. The QtXmlcontains classes for working with XML files. This module provides implementation for both SAX and DOM APIs. The QtSvg module provides classes for displaying the contents of SVG files. Scalable Vector Graphics (SVG) is a language for describing two-dimensional graphics and graphical applications in XML. The QtOpenGL module is used for rendering 3D and 2D graphics using the OpenGL library. The module enables seamless integration of the Qt GUI library and the OpenGL library. The QtSql module provides classes for working with databases.

**Coding**

import speech\_recognition as sr  
  
  
recognizer = sr.Recognizer()  
  
  
''' recording the sound '''  
  
  
with sr.Microphone() as source:  
  
    print("Adjusting noise ")  
  
    recognizer.adjust\_for\_am bient\_noise(source, duration=1)  
  
    print("Recording for 4 seconds")  
  
    recorded\_audio = recognizer.listen(source, timeout=4)  
  
    print("Done recording")  
  
  
''' Recorgnizing the Audio '''  
  
try:  
  
    print("Recognizing the text")  
  
    text = recognizer.recognize\_google(  
  
            recorded\_audio,   
  
            language="en-US"  
  
        )  
  
    print("Decoded Text : {}".format(text))  
  
  
except Exception as ex:  
  
    print(ex)

**CONCLUISION**

Speech Recognition has become very important in today’s world. With the advancements in technology and improvements in recognition algorithms, speech has become one of the primary source of input for many applications. Speech is the most efficient and natural way of communication. So, it is intuitive that speech recognition systems have found applications in various fields.

Interactive Voice Response (IVR) systems are one of the prominent systems that have  
a huge potential for use of voice signals as input to the system. With this in mind, we  
presented an idea for the development of an IVR system with Automatic Speech Recognition (ASR). The initial objective of the project was to develop a system capable of recognizing voice signals in Nepali Language input to the IVR system. Throughout the course of the development phase, various limitations and obstacles were encountered which prompted us to develop the system capable of recognizing words corresponding to the digits of the Nepali Language. For this, we researched on various methods of speech recognition and used the findings of these researches to develop the system.  
The project was implemented by using algorithms like Noise Reduction, Voice Activity  
Detection, MFCC Feature Extraction, Hidden Markov Model and Recurrent Neural Network.

The overall accuracy of the system while using HMM was around 70 percentage and while using RNN was around 80 percentage. The greater accuracy of RNN is due to the fact that, RNN do not make Markov assumptions and can account for long term dependencies when modeling natural language and due to the the greater representational power of neural networks and their ability to perform intelligent smoothing by taking into account syntactic and semantic features. Though the accuracy seems to be a bit less, the accuracy is good compared to the fact that we had such less data set available. With proper amount of data set available the project can get much higher accuracy and can be implemented.

**LIMITATIONS AND FURTHER WORKS**

Speech Recognition has been a very interesting field in research and technology.Many  
technical teams around the globe are working together to get the satisfactory result.Several researches are ongoing in this field and due to advancement in technology and efficient new models it has made possible to make further enhancement in this field to get more accurate results.Like many other projects our project has also got limitations and the enhancements that can be made in future.

**Limitations**

Some of the particular field of limitations are:

**Narrow Recognition Domain:** Currently the IVR system with speech recognition system works on very narrow domain. Because of time limitation and difficulty in collecting the data samples at present we are focused on using only Nepali numbers from zero to nine in the system.The accuracy level of recognition is dependent on available number of training samples but due to unavailability of training data only  
fewer data are being trained and recognized.

**Offline operation:** Another limitation of the current system is that it is designed only for the offline operation i.e. available only on desktop environment but not on the web.

**Narrow application domain:** Our present system is focused only in implementation speech recognition on automating a simple task in desktop environment.

**Future Enhancements**

The potential enhancements that can be made to the system are discussed below:  
1. By increasing the training data samples using effective data collection mechanism the  
domain of recognition can be increased.  
2. The system may be enhanced to make work for online mode by integrating it in web  
applications.  
3. The system can be enhanced to apply on the real time applications using telephone.  
For this further research on particular domain is necessary.

**REFERENCES**

[1] Christopher Olah. Understanding LSTM Networks. http://colah.github.io/posts/  
2015-08-Understanding-LSTMs/

.  
[2] Andrej Karpathy. The Unreasonable Effectiveness of Recurrent Neural Networks. <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>.

[3] Md Salam, Dzulkifli Mohamad, and Sheikh Salleh. Malay isolated speech recognition  
using neural network: A work in finding number of hidden nodes and learning  
parameters. *The International Arab Journal of Information Technology*, 8, 2011.

[4] Hidden Markov model. In *Wikipedia*. https://en.wikipedia.org/wiki/Hidden\_  
Markov\_model.

[5] Markov model. In *Wikipedia*. <https://en.wikipedia.org/wiki/Markov_model>.

[6] Dr. Jason Brownlee. Machine Learning Mastery Blog Series. http://  
machinelearningmastery.com/blog/.

[7] James Robert. Pydubs. <https://github.com/jiaaro/pydub>

[8] James Lyons. Spectral Subtraction Demo. https://gist.github.com/jameslyons/554325efb2c05da15b31.

[9] The Scipy Community. Numpy Documentation. https://docs.scipy.org/doc/  
numpy-1.13.0/reference/.

[10] James Lyons. Mel Frequency Cepstral Coefficient (MFCC) Tutorial.  
http://www.practicalcryptography.com/miscellaneous/machine-learning/  
guide-mel-frequency-cepstral-coefficients-mfccs/