A

Major Project Report

on

SPEECH TO TEXT FOR LOCAL ETHNIC LANGUAGE: AHIRANI

Submitted in Partial Fulfillment of
the Requirements for the Final Year
of

Bachelor of Engineering

in

Computer Engineering

to

Kavayitri Bahinabai Chaudhari
North Maharashtra University, Jalgaon

Submitted by

Gaurav R. Baviskar
Mahesh P. Kanhere
Mohit M. Chaudhari
Pooja M. Shirude

Under the Guidance of

Mr.Pravin K. Patil

DEPARTMENT OF COMPUTER ENGINEERING
SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY,
BAMBHORI, JALGAON - 425 001 (MS)
2022 - 2023

SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY,
BAMBHORI, JALGAON - 425 001 (MS)
DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the major project entitled *SPEECH TO TEXT FOR LOCAL ETHNIC LANGUAGE: AHIRANI*, submitted by

Gaurav R. Baviskar
Mahesh P. Kanhere
Mohit M. Chaudhari
Pooja M. Shirude

in partial fulfillment of the degree of *Bachelor of Engineering* in *Computer Engineering* has been satisfactorily carried out under my guidance as per the requirement of Kavayitri Bahinabai Chaudhari North Maharashtra University, Jalgaon.

Date: March 31, 2023
Place: Jalgaon

Mr.Pravin K. Patil
Guide

Dr. Manoj E. Patil
Head

Dr. G.K.Patnaik
Principal

Acknowledgements

We would like to express our deep gratitude and sincere thanks to all who helped us in completing this project report successfully. Many thanks to almighty God who gave us the strength to do this. Our sincere thanks to Principal Dr. Girish. K. Patnaik for providing the facilities to complete this Project report. We would like to express our gratitude and appreciation to all those who gave us the possibility to complete this report. A special thanks to Prof. Dr. Manoj E. Patil, Head of the Department, whose help, stimulating suggestions and encouragement, helped us in writing this report. We would also like to thank Mr. Pravin K. Patil, Project Guide, who has given his full effort in guiding us and achieving the goal as well as his encouragement to maintain the progress in track. We are also sincerely thankful to Mr. Akash D. Waghmare, Incharge of Project, for his valuable suggestions and guidance. We would also like to appreciate the guidance given by other supervisor that has improved our presentation skills by their comments and tips. Last but not the least, we are extremely thankful to our parents and friends without whom it could not reach its successful completion.

Gaurav R. Baviskar
Mahesh P. Kanhere
Mohit M. Chaudhari
Pooja M. Shirude

Contents

List of Tables

List of Figures

Abstract

This report represents an overview for Automatic Speech Recognition (ASR) for our local ethnic language Ahirani which is a commonly spoken language in the Khandesh region of Maharashtra State of India. Speech is the most powerful way of communication with which human beings express their thoughts and feelings through different languages. The features of speech differs with each language. However, even while communicating in the same language, the pace and the dialect varies with each person. This creates difficulty in understanding the conveyed message for some people. Sometimes lengthy speeches are also quite difficult to follow due to reasons such as different pronunciation, pace and so on. Speech recognition which is an inter disciplinary field of computational linguistics aids in developing technologies that empowers the recognition and translation of speech into text. Text summarization extracts the utmost important information from a source which is a text and provides the adequate summary of the same. The research work presented in this report describes an easy and effective method for speech recognition. Extensive experimentation is performed to validate the efficiency of the proposed method.

Keywords:-Ahirani, Automatic Speech Recognition (ASR), HMM, HTK, Isolated Word ASR, Mel Frequency Cepstral Coefficient (MFCC), Speaker Independent.

Chapter 1

Introduction

Speech is the most important part of communication between human beings. Though there are different means to express our thoughts and feeling, speech is considered as the main medium for communication.Speech recognition is the process of making a machine recognize the speech of different people based on certain words or phrases. It is a technology that allows a computer to identify the words that a person speaks into a microphone or telephone. Speech recognition can be defined as the process of converting an acoustic signal, captured by a microphone or a telephone, to a set of words [1][2]. Automatic speech recognition (ASR) is one of the fastest growing areas of engineering and technology. Automatic speech recognition systems are developed for English and other major languages spoken in developed countries. Automatic speech recognition systems are under development for Indian languages such as Hindi, Tamil, Telugu, Bengali, Assamese and Marathi. Spoken languages like Ahirani are not explored till now.This work is an attempt to initiate the work on designing and developing a speech recognition system for Ahirani. It is one of the most common language spoken in Khandesh. Khandesh region mainly constitutes Dhulia, Jalgaon and Nandurbar districts. Automatic speech recognition systems have been implemented using various toolkits and software. Most commonly used amongst them are the Hidden Markov Model ToolKit, Sphinx Toolkit, ISIP Production System, Julius Open-Source Large Vocabulary CSR Engine, HMM Toolbox for Matlab etc. Among all these tools the HTK toolkit is the most popularly used tool to design ASR systems. Since it is used in building and manipulating hidden Markov Models it has applications in other research areas as well. HTK is well documented and provides guided tutorials for its use.

In this chapter section 1.1 presents Background, section 1.2 presents Motivation,section 1.3 presents Problem Definition, section 1.4 presents Scope , Objectives are presented in section 1.5 , Selection of Life cycle model is presented in section 1.6, Organization of Report is described in section 1.7, and Summary is described in section 1.8

1.1 Background

In the major part of the Maharashtra, namely Khandesh which consists of almost three districts- Jalgaon, Dhule, Nandurbar, after having significant number of speakers. Still the language is lacking the digital presence and its use by non-native speakers. Also the literature which is present till date is not getting the enough exposure by the audience due to lack of reach due to which many of its part is on it's way to extinction.

1.2 Motivation

At present, due to its versatile applications, speech recognition is the most promising field of research. Our daily life activities, like mobile applications, weather forecasting, agriculture, healthcare etc. involves speech recognition.The automatic speech recognition systems are developed for English and other major languages spoken in developed countries.However automatic speech recognition are under development for Indian languages like Hindi, Telugu, Marathi, Bengali, Assameses, etc. but no one provides efficient solution for Ahirani language. There is no any automatic speech recognition system i.e, speech to text or text to speech for Ahirani language. The lack of effective Ahirani speech recognition system and its local relevance has motivated the authors to develop such speech to text system for Ahirani language.

1.3 Problem Definition

The problem is that there is no any Automatic Speech Recogniser, speech-to-text or text-to-speech for our local ethnic language Ahirani.
And the only chance for any language to survive till last is having its digital presence is must in todays world.

Although having the significant number of speakers Ahirani or similar ethnic languages are getting deprived day by day in their own land due to absence of ease in accessibility at digital platforms.

Due to Civilisation, Migration there is continues reduction reduction in number of speakers.

No unified portal of Ahirani Language

1.4 Scope

Ahirani Speech to text converter can be used at various places such as ATM, Printing Machines etc.

It can help to maximize the Human-machine interaction by various means.

Further dataset can be extended and can achieve maximum accuracy for better performance.

By creating the new corpus similar systems can be made for variety of local/rural languages.

1.5 Objectives

We aimed to develop the Automatic Speech Recognition (ASR) system for local language Ahirani. For the same we will be building our own unique corpus for Ahirani language and planed to make Machine Learning Model well as the data-set open source for future research and use. Not only limited to this we will be giving Ahirani literature a strong digital presence, to critically review the literature and made it accessible to all intended users.Developing this ASR will help to create an ease between the people and various digital platforms.

1.6 Selection of Life cycle model

The software development life cycle model selected for this project is the Waterfall Model. Waterfall approach was the first SDLC (Software Development Life Cycle) Model to be widely used in software engineering to ensure success of the project. It was developed by Winston W. Royce in 1970. In "The Waterfall" approach, the whole process of software development is divided into separate phases, typically the outcome of one phase acts as the input for the next phase sequentially. All the phases are cascaded to each other in which progress is seen as owing steadily downwards (like a waterfall) through the phases. Requirements for this project are well documented and fixed up.

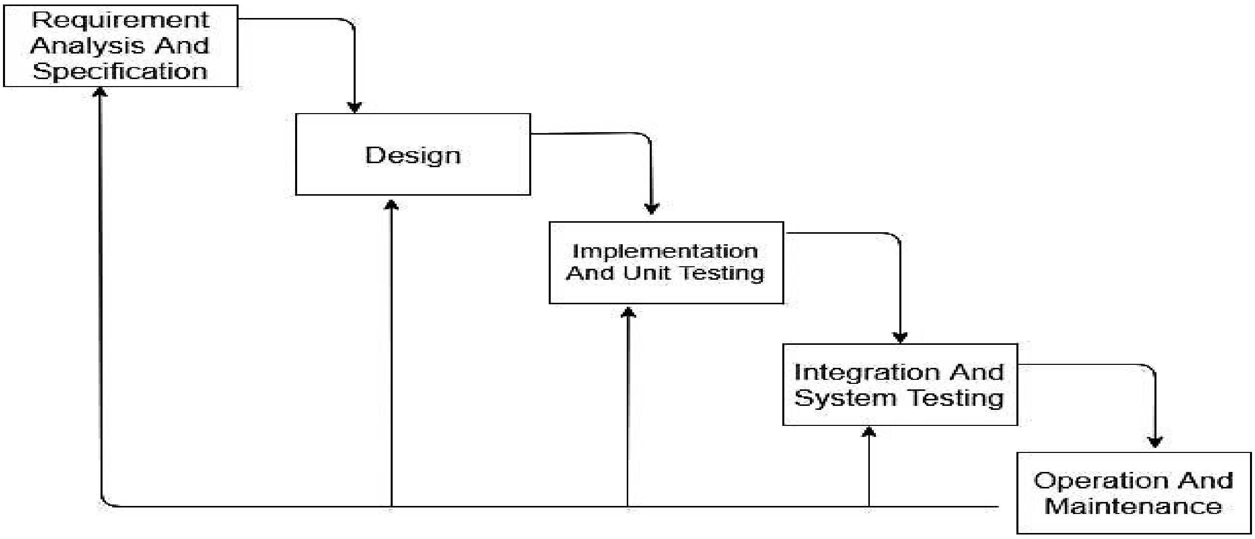

Requirement Analysis And Specification
Design
Implementation And Unit Testing
Integration And System Testing
Operation And Maintenance

Figure 1.1: Waterfall Model

Waterfall Model is best suited model for this project.

1. Because requirements are easily understandable and defined
2. We can define requirement in early stage of development
3. User involvement in all phases is not necessary
4. Limited user's participation.

1.7 Organization Of Report

- Chapter 1: Introduction this section will provide general information of this project, the Project Definition, Scope, the Objectives to be achieved, selection of Life Cycle Model on the basis of analysis of gathered requirements, etc

- Chapter 2: Project Planning and Management This section includes the Feasibility Study that summarizes results of the analysis and evaluations conducted to review the propose solution and for the purpose of identifying if the project is really feasible, cost-effective and profitable. Risk analysis could include legal/contractual risks, environmental risks, revenue risks, project management risks, regulatory risks etc. A project schedule communicates what tasks need to get done and which organizational resources will be allocated to complete those tasks in what time frame.

- Chapter 3: Analysis This chapter focuses on the gathering of requirements/data and their identification. The Software Requirements Specification (SRS) document lists sufficient and necessary requirements, functional and non-functional requirements, their operating environment, product features, etc. For the project development.

- Chapter 4: Design This chapter contains important aspects like the System Architecture, Data Flow Diagrams (DFDs), UML diagrams which contains the Blueprints of the system. They highlight the architecture, flow and sequential execution of the events.

- Chapter 5: Conclusion and Future Scope The Conclusions section sums up the key points of the discussion/Project, the essential features of the design, or the significant outcomes of the investigation. Also, bibliography, index and appendix.

1.8 Summary

This project ”Speech to text converter for Ahirani Language” is used to convert the audio into textual format. This model uses the custom made corpus specifically for Ahirani Language. This project model helps to provide digital surveillance for the local ethnic language. It provides unified portal for it’s literature to be accessible.

Chapter 2

Project Planning And Management

Planning is the process of gathering and interpreting facts and diagnosing problems and the facts to improve the system. Project planning can be done manually or by the use of project management software. The organization of the chapter is as follows:- Section 2.1 Feasibility study and in it, we will see 2.1.1 Economic Feasibility, 2.1.2 Operational Feasibility, 2.1.3 Technical Feasibility, then we will see 2.2 Risk Analysis and 2.3 Project Scheduling and 2.4 Effort Allocation and at last we will see short Summary of this chapter in 2.5.

2.1 Feasibility Study

The feasibility study is carried out to test whether the proposed system is worth being implemented. Feasibility study is a test of system proposed regarding its work ability, its impact on the organization ability to meet user needs and effective use of resources. The key consideration involve in the feasibility study are:

- Economic Feasibility
- Operational Feasibility
- Technical Feasibility

2.1.1 Economical Feasibility

This is the main factor in the feasibility study. When product is economically affordable then it can be used. So project must be cost saving. Establishing the cost effectiveness of the proposed system. The project relies on browser-based interface for end user inputs. Since the interface is completely browser based it doesn't require bandwidth allocation and thus reduces the financial aspect of project implementation. The project requires some software and tools which follow freeware software standards and the dataset we are using is our own dataset.

2.1.2 Operational Feasibility

Operational feasibility is the ability to utilize, support and perform the necessary tasks of a system or program. It includes everyone who creates, operates or uses the system Provide summary statistical information without disclosing individual's confidential data. This makes the system operationally feasible. The System we had developed is speech to text for Ahirani Language Keeping this thing in mind we are developing a browser-based client which would be freely available for people thus this will be directly beneficial for our society.

2.1.3 Technical Feasibility

This assessment is based on an outline design of system requirements, to determine whether the company has the technical expertise to handle completion of the project. Technical feasibility involves the evaluation of the hardware, software, and other technical requirements of the proposed system. This portal is developed using development technologies such as Python, Html, CSS, and JS . All the required hardware and software are easily available in the market. Hence the portal is technically feasible. There is no more hardware required other than personal system.

2.2 Risk Analysis

Risk analysis and management are a series of steps that help a software team to understand and manage uncertainty. Many problems can plague a software project. A risk is a potential problem might happen, it might not. But, regardless of the outcome, it is really a good idea to identify it, assess its probability of occurrence, estimate its impact, and establish a contingency plan should the problem actually occur.

2.3 Project Scheduling

Software project scheduling is an activity that distributes estimated effort across the planned project duration by allocating the effort to specific software engineering tasks. It is important to note, however, that the schedule evolves over time. During early stages of project planning, a macroscopic schedule is developed. The schedule identities all major software engineering activities and the product functions to which they are applied. As the project gets under way, each entry on the macroscopic schedule is refined into a detailed schedule.

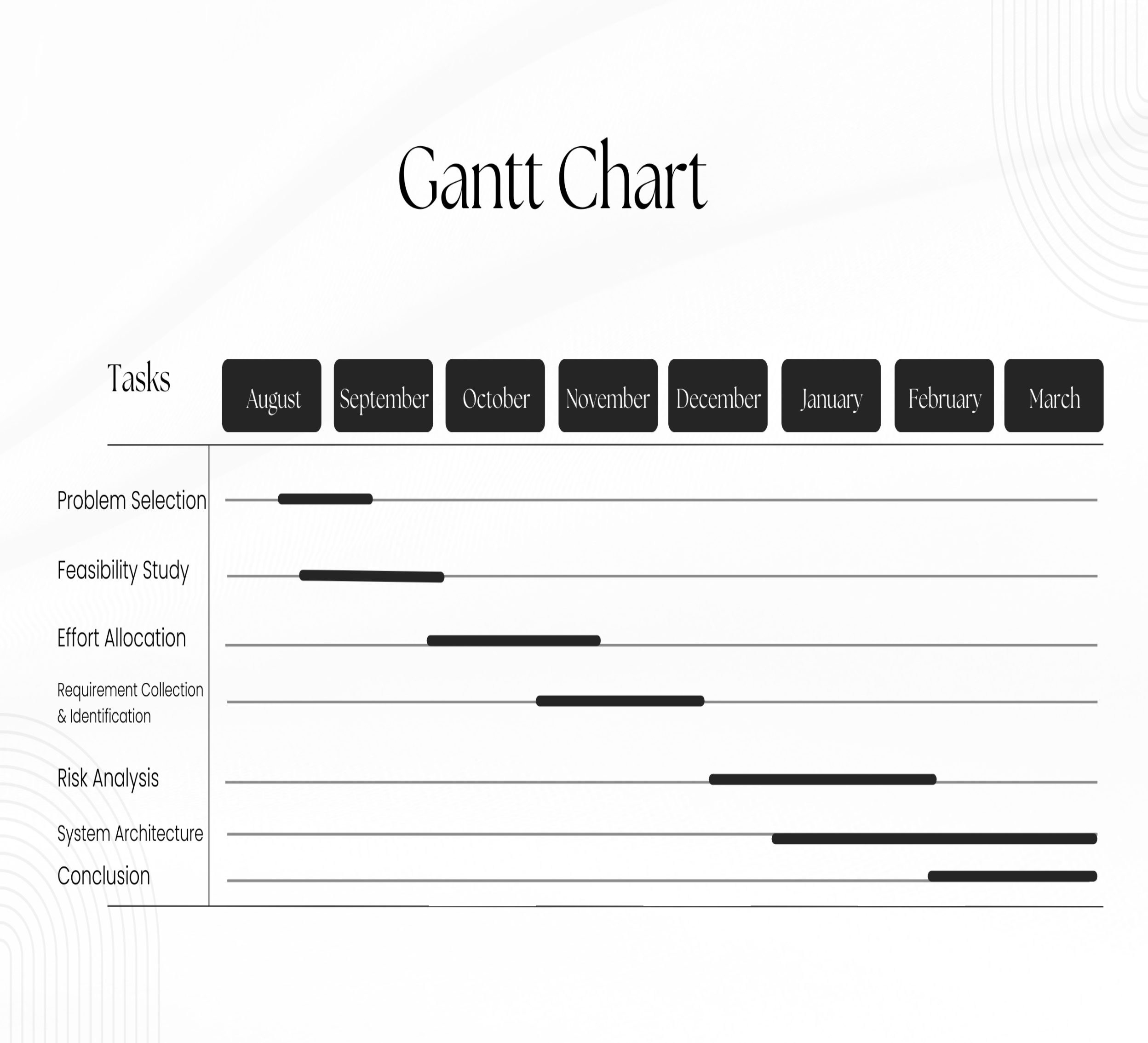


Figure 2.1: project Scheduling (Gantt chart)

2.4 Effort Allocation

Project means team work, Project is developed by combination of effort of team. So whole project is divided into modules and number of modules is allotted to team members.

Task	Pooja Shirude	Mohit Chaudhari	Mahesh Kanhere	Gaurav Baviskar
Discussing Current situation		✓		✓
Discussing the solution	✓	✓	✓	✓
Documentation of the solution	✓	✓		
Study of Tech Stack required	✓	✓	✓	✓
Selection of life cycle model	✓	✓	✓	✓
Planning And management		✓	✓	✓
Analysis And Design	✓		✓	✓

Table 2.1: Effort Allocation

2.5 Summary

The Project is henceforth, feasible to use as its being developed primarily Speech to text conversion of local ethnic language any speaker can use the feature to perform conversion of speech to text.

Chapter 3

Analysis

The Analysis Phase is where the project lifecycle begins. It helps the software engineers to better understand the problem they will work to solve. The analysis is also part of the project where we identify the overall direction that the project will take through the creation of the project strategy documents. In the analysis phase, end user requirements are analyzed and project goals converted into the defined system functions that take project intends to develop.

Section 3.1 presents the Requirement Collection and Identification. Software Requirement Specification is described in Section 3.2. Section 3.1 presents the requirement Collection,3.1.1 Reqiurement Identification 3.1.2. Section, section 3.2.1 presents product feature , 3.2.2 presents Operating Environment,3.2.3 Assumption, 3.2.4. presents , Functional Requirements,3.2.5 presents Non-Functional Requirements,3.2.6 preseents External Interfaces and 3.3 presents summary of this chapter

3.1 Requirement Collection and Identification

Requirement Collection is the main phase of the Analysis phase because this is when the project team begins to understand what the end user wants from the project.

3.1.1 Requirement Collection

- Research Papers based on Ahirani ASR.
- Already existing Dataset.
- Available Pre-trained model
- Different libraries to perform intended task

3.1.2 Requirement Identification

Requirement Collection and Identification refers to the process of collecting the requirement of the user through various methods. Analyzing them and documenting them, for creating a software requirement specification. And developing a product that meets the requirement of the user and gradually solves all the problems. Requirement analysis helps organizations to determine the actual needs of stakeholders. At the same time, it enables the development team to communicate with stakeholders in a language they understand (like charts, models, flow-charts,) instead of pages of text. Requirements analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design. The task of communicating with customers and users to determine what their requirements are. This is sometimes also called requirements gathering, determining whether the stated requirements are unclear, incomplete, ambiguous, or contradictory, and then resolving these issues. Requirements analysis is a team effort that demands a combination of hardware, software, and human factors engineering expertise as well as skills in dealing with people.

3.2 Software Requirements Specifications (SRS)

It provides requirements, needs of project and those things which help to complete project. System requirement describe a system from a technical perspective, which describe the essential characteristics of the hardware and software that will meet those needs. It should specify the capabilities, capacities and characteristics of the system in both qualitative and quantitative terms.

3.2.1 Product Feature

- User can process his/her speech signal into text
- System can be integrated with other applications for better implementations: such as ATM machines, Passbook printing machine etc.
- User can copy or download the output generated in textual format
- Portal to share literature available in Ahirani language

3.2.2 Operating Environment

The software will operate within the following environment:

- Operating System: Windows 10 or later/Linux/MacOS
- Any system with at least 4 GB RAM
- Any browser supporting HTML5 and JavaScript
- System with processor Intel Pentium 4 or later
- Google chrome recommended web browser for use of website

3.2.3 Assumption

This is external factor for which the intervention is not responsible, but that are very important for the realization of the results, the project purpose and the overall objective. Some assumption are as follows:

- Noise in the Speech signals can affect the output
- Corpus created is error free
- All the elements are in the working process through out the project development life cycle
- Team will have to use Open Source tools and resources.

- Noise in the Speech signals can affect the output
- Corpus created is error free
- All the elements are in the working process through out the project development life cycle
- Team will have to use Open Source tools and resources.

3.2.4 Functional requirements

Functional requirements are the functions which are expected from the software or platform. Functional requirements along with requirement analysis help identify missing requirements. They help clearly define the expected system service and behavior

- Input Data: User should provide input data in required format
- output Data: Output is text, based on the input given by the user
- Download Data: User will get the option to download data in .txt format

3.2.5 Non-Functional Requirements

Non-functional Requirement is mostly quality requirement. That stipulates how well the portal does, what it has to do. Other than functional requirements in practice, this wouldn entail detail analysis of issues such as availability, security, usability and maintainability. Non-functional requirements are as follows:

- Response Time : The purpose of identifying response time is to highlight perform of new website .Normal customer response shall be less than 5 sec for all website
- Workload : The new website must support 100 people at any given time and it might double traffic volume in upcoming days.
- Security: The main security concern is for users account that's why Login mechanism should be used to avoid unauthorized access.
- Safety: Information transmission should be securely transmitted to server without any changes in information.
- Availability: If the internet service gets disrupted while sending information can be send again for verification.
- Usability: As the system is easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its states.

3.2.6 External Interfaces (User, Hardware, Software, Communication)

- *Hardware Interface*

It is web based product. The hardware on which it resides will be any computer can have internet. The hardware requirement includes a system with following configurations:

- Processor: Intel Pentium or above
- RAM: 1 GB
- Input device: Standard Microphone
- Output device : High Resolution Monitor

- *Software Interface*

Home Page : The homepage is design with colorful interface and intuitive layout for easy navigation so that even non-technical person can easily navigate through the website. Also the homepage will be user friendly and on homepage user will provide the input.

Upload audio: Here user can upload speech signal to process

Convert: Here user can convert his/her input audio into text format

Download : User can download Textual content generated here in the form of '.txt' format.

Front-end: HTML, CSS, JS

Back-end: Python

Server: Flask (localhost)

- *Communication Interfaces*

As a part of its core functionalities this product will require HTTP OR HTTPS communication interface with client device. It will also require to communicate with SQL DATABASE.

3.3 Summary

The Project is henceforth, feasible to use as its being developed primarily Speech to text conversion of local ethnic language any speaker can use the feature to perform various kinds of routine as well as multinational tasks.

Chapter 4

Design

System design provides the understanding and procedural details necessary for implementing the system. Design is an activity concerned with making major decisions, often of a structural nature. Design builds coherent, well planned representations of programs that concentrate on the interrelationships of parts at the higher level and the logical operations involved at the lower levels. Software design is the first of the three technical activities designs, coding and test which are required to build and verify the software. This model, known as the design model, is assessed for quality and reviewed before a code is generated and tests are conducted.

4.1 Section presents System architecture, data flow diagram is presented in Section 4.2, Section 4.3 presents UML diagrams of the proposed system. Finally, the Summary is presented in the Last Section.

4.1 System Architecture

The System Architecture provides the details of how the components or modules are integrated. Figure 2 is indicating the system architecture of the speech to text model. This architecture will give complete description of input and outputs of each process.

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture, collectively these are called architecture description languages.

The purpose of system architecture activities is to define a comprehensive solution based on principles, concepts, and properties logically related to and consistent with each other.

The developed speech recognition system architecture is shown in figure 2 It consists of two modules, training module and testing module. Training module generates the system model which is to be used during testing. The various phases used during ASR are:

Preprocessing: Speech-signal is an analog waveform which cannot be directly processed by digital systems. Hence preprocessing is done to transform the input speech into a form that can be processed by recognizer. To achieve this, firstly the speech-input is digitized. The digitized (sampled) speech-signal is then processed through the first-order filters to spectrally flatten the signal. This process, known as pre-emphasis, increases the magnitude of higher frequencies with respect to the magnitude of lower frequencies. The next step is to block the speech-signal into the frames.

Feature Extraction: The goal of feature extraction is to find a set of properties of an utterance that have acoustic correlations to the speech-signal, that is parameters that can some how be computed or estimated through processing of the signal waveform. Such parameters are termed as features. The feature extraction process is expected to discard irrelevant information to the task while keeping the useful one. It includes the process of measuring some important characteristic of the signal such as energy or frequency response (i.e. signal measurement),augmenting these measurements with some perceptually meaningful derived measurements (i.e. signal parameterization), and statically conditioning these numbers to form observation vectors.

Model Generation: The model is generated using various approaches such as Hidden Markov Model (HMM) (Huang et al., 1990), Artificial Neural Networks (ANN) (Wilinski et al., 1998), Dynamic Bayesian Networks (DBN) (Deng, 2006), Support Vector Machine (SVM) (Guo and Li,2003) and hybrid methods (i.e. combination of two or more approaches). Hidden Markov model has been used in some form or another in virtually every state-of-the-art speech and speaker recognition system (Aggarwal and Dave, 2010).

Pattern Classifier: Pattern classifier component recognizes the test samples based on the acoustic properties of word. The classification problem can be stated as finding the most probable sequence of words W given the acoustic input O (Jurafsky and Martin, 2009), which is computed as: Pattern classifier component recognizes the test samples based on the acoustic properties of word. The classification problem can be stated as finding the most probable sequence of words W given the acoustic input O (Jurafsky and Martin, 2009), which is computed as:

$$P(W|O) = \frac{P(O|W).P(W)}{P(O)}$$

Given an acoustic observation sequence O, classifier finds the sequence W of words which maximizes the probability P(O —W).P(W) . The quantity P(W) , is the prior probability of the word which is estimated by the language model. P(O —W) is the observation likelihood, called as acoustic model.

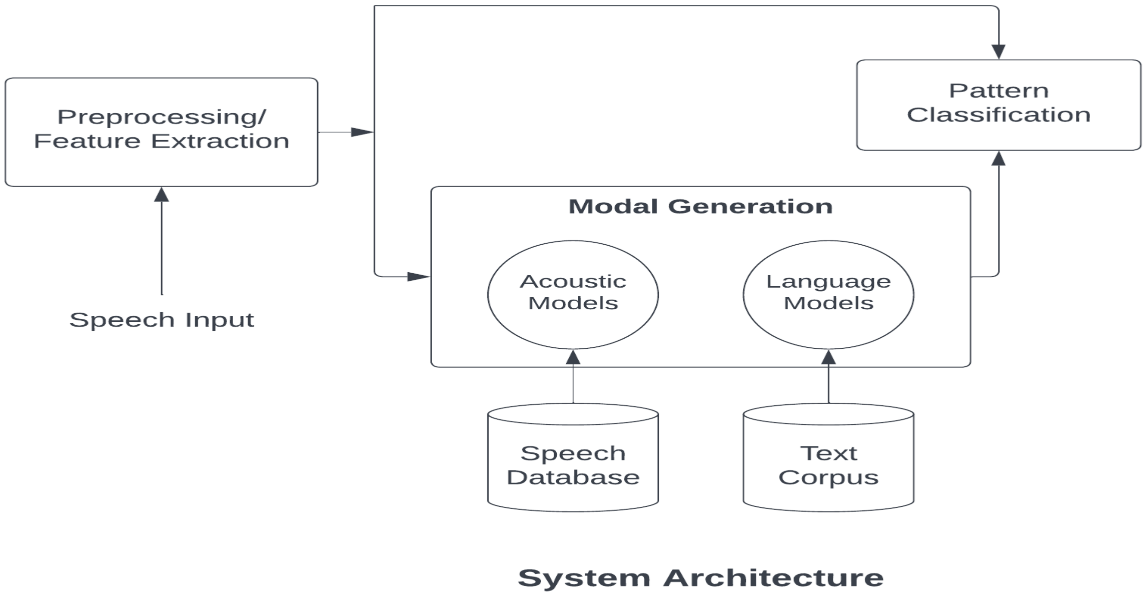


Figure 4.1: System Architecture

4.2 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the 'flow' of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored.

4.2.1 Level 0 DFD

Level 0 contains one input and one output. The system provides information to the user means system is input and the user is output. Figure shows Level 0 DFD of project.

It is also known as a context diagram. It's designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows.

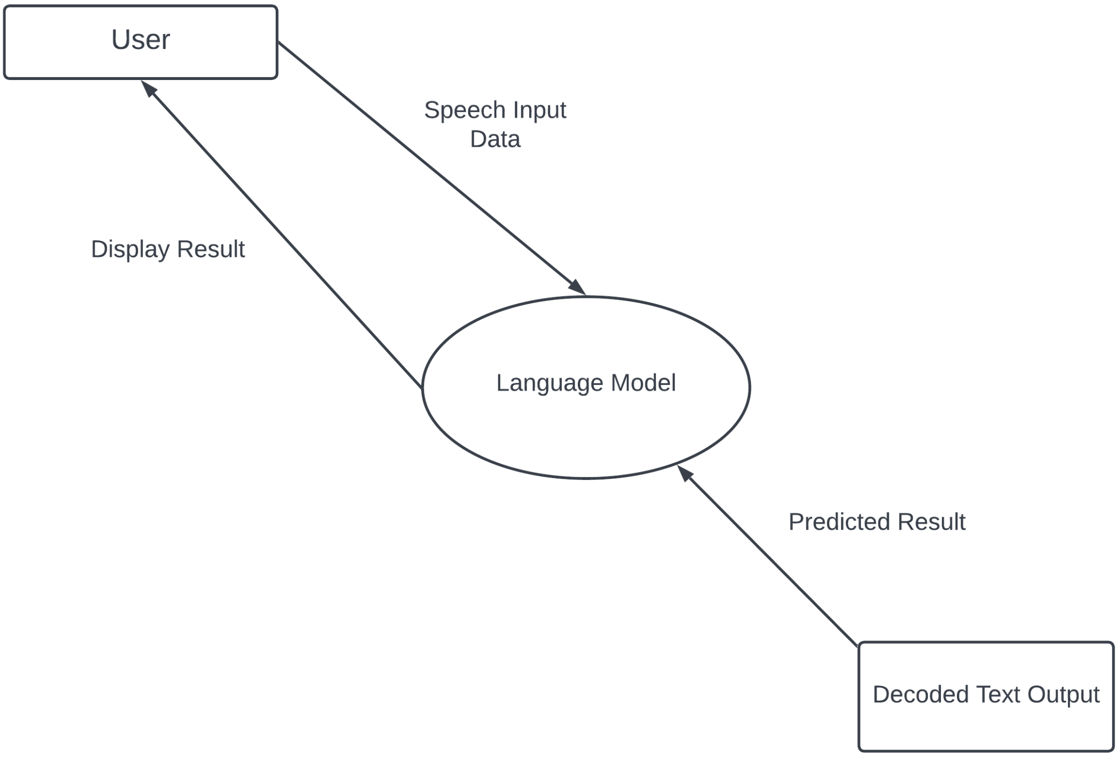


Figure 4.2: Level Zero Data Flow Diagram

4.2.2 Level 1 DFD

A level 1 DFD notates each of the main sub-processes that together form the complete system. We can think of a level 1 DFD as an ”exploded view” of the context diagram. Figure shows Level 1 DFD of the project. In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes.The Level 0 DFD is broken down into more specific, Level 1 DFD. Level 1 DFD depicts basic modules in the system and flow of data among various modules. Level 1 DFD also mentions basic processes and sources of information. It provides a more detailed view of the Context Level Diagram.Here, the main functions carried out by the system are highlighted as we break into its sub-processes.

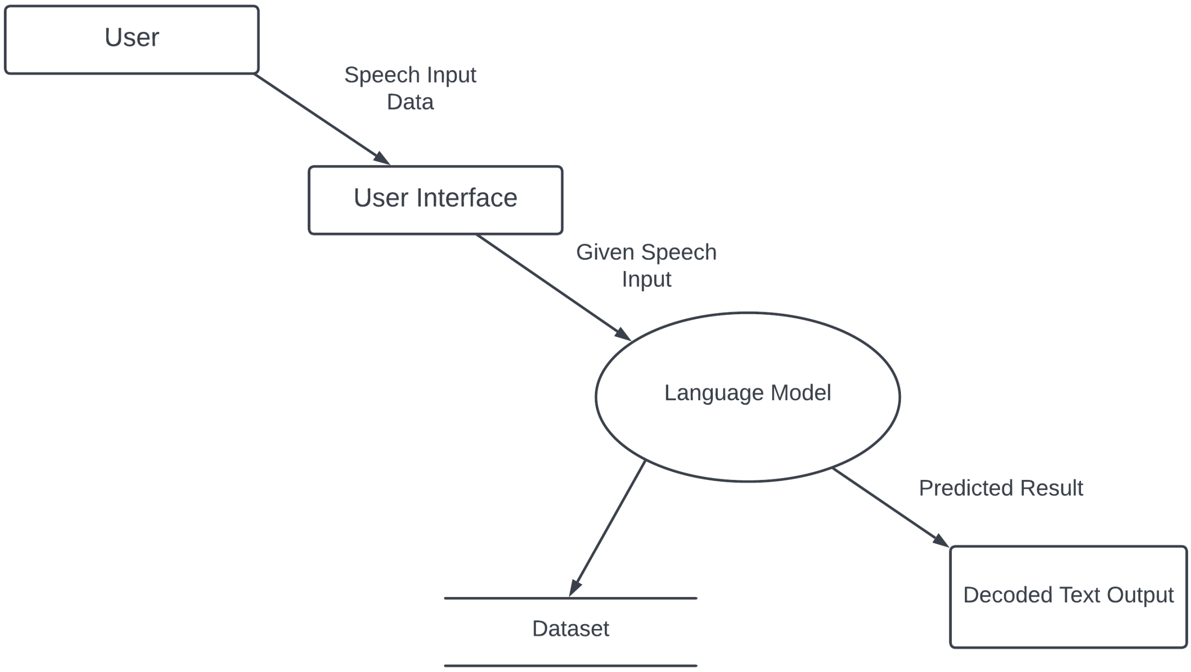


Figure 4.3: Level 1 Data Flow Diagram

4.2.3 Level 2 DFD

A level 2 data flow diagram offers a more detailed look at the processes that make up an information system than a level 1 DFD does. It can be used to plan or record the specific makeup of a system. Figure shows Level 2 DFD of project.

Figure shown below is the level 2 DFD . The User has required to input the audio in order to convert into the textual format. System will process the audio generate the output in textual format.

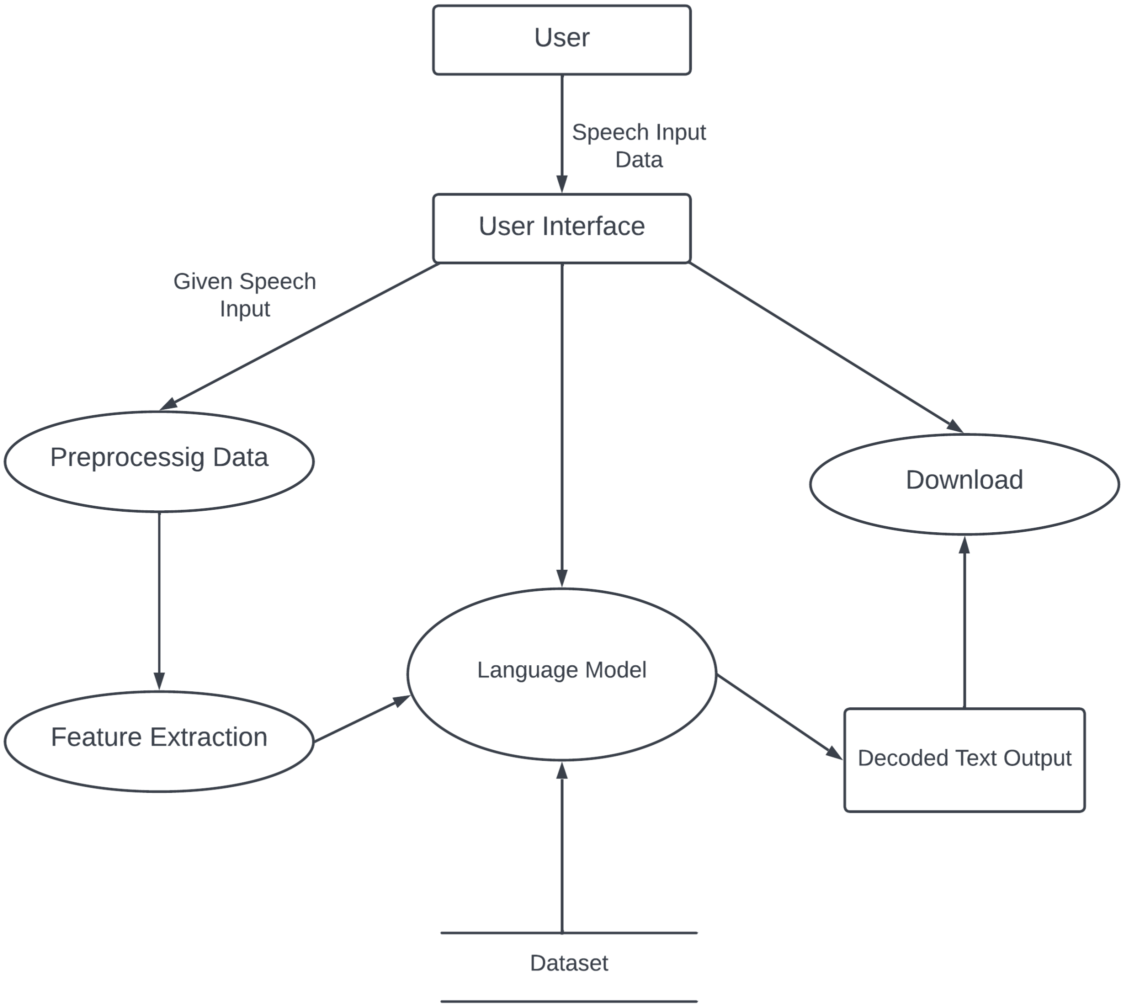


Figure 4.4: Level 2 Data flow diagram

4.3 UML Diagrams

The UML is a language for:

- Visualizing:-The structures which are transient can be represented using the UML
- Specifying:-The UML addresses the specification of all the important analysis, de- sign and implementation decisions that must be made in developing and deploying a software-intensive system
- Constructing:-The UML is not a visual programming language, but its models can be directly connected to a variety of programming languages

- Documenting:-The UML addresses the documentation of a system's architecture and all of its details.

4.3.1 Use Case Diagram

Use case diagram shows the interaction between Use case which represents system functionality and actor which represent the people or system.

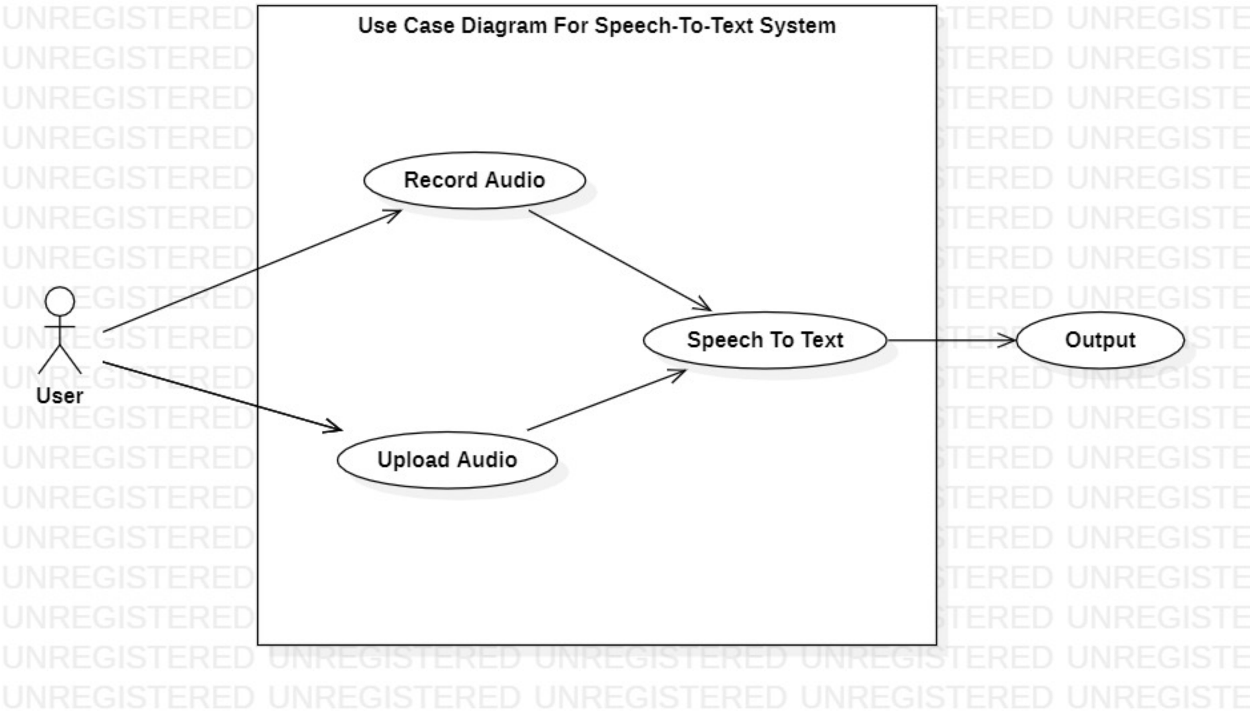


Figure 4.5: Use Case Diagram

4.3.2 Sequence Diagram

Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. Figure shows sequence diagram for Use cases

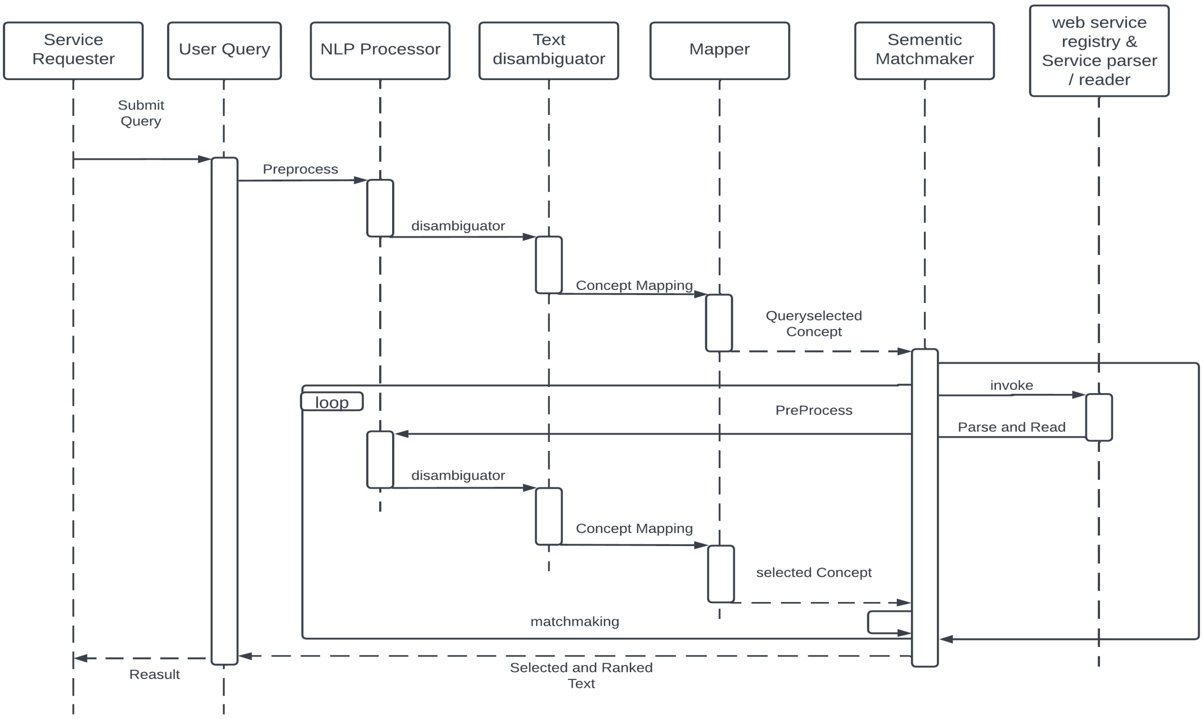


Figure 4.6: Sequence Diagram

4.3.3 Collaboration Diagram

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system. Figure, shows Collaboration diagram for Use cases.

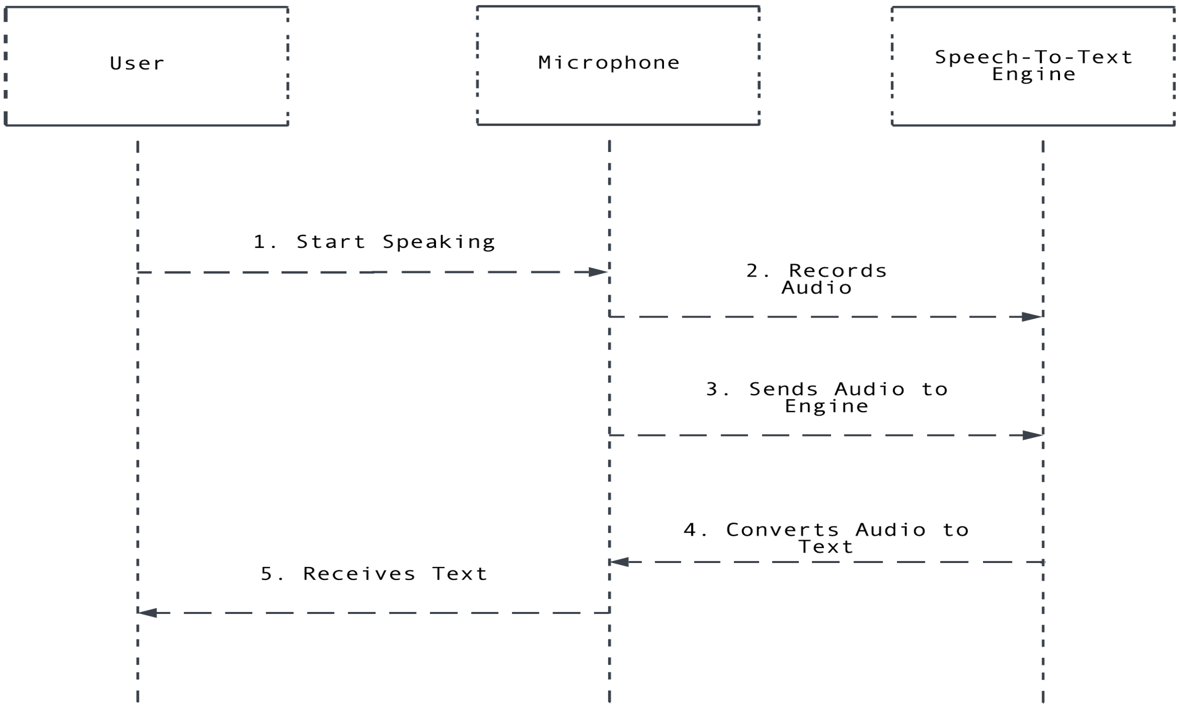


Figure 4.7: Collaboration Diagram For Sequence Diagram- Showing Prediction Graph

4.3.4 Class Diagram

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. Figure shows class diagram for this project.

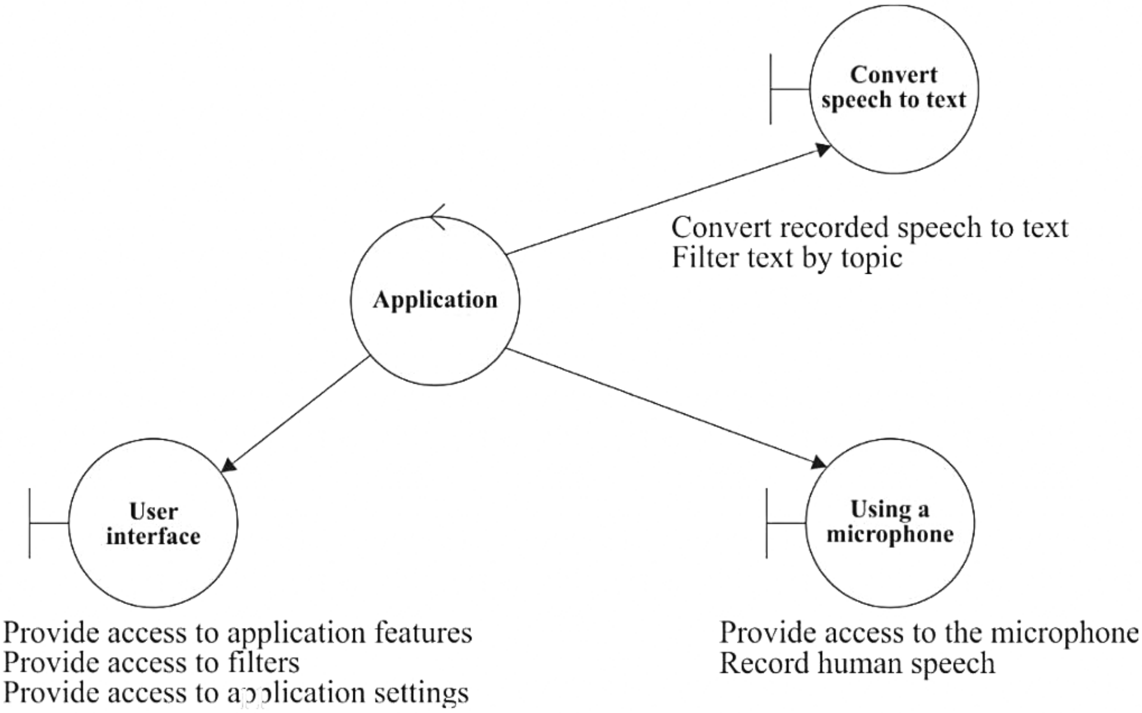

Convert speech to text
Convert recorded speech to text
Filter text by topic
Application
User interface
Using a microphone
Provide access to application features
Provide access to filters
Provide access to application settings
Provide access to the microphone
Record human speech

Figure 4.8: Class Diagram

4.3.5 State Chart Diagram

A state chart diagram, also known as a state machine diagram or state chart diagram, is an illustration of the states an object can attain as well as the transitions between those states in the Unified Modelling Language (UML). Figure shows State chart Diagram.

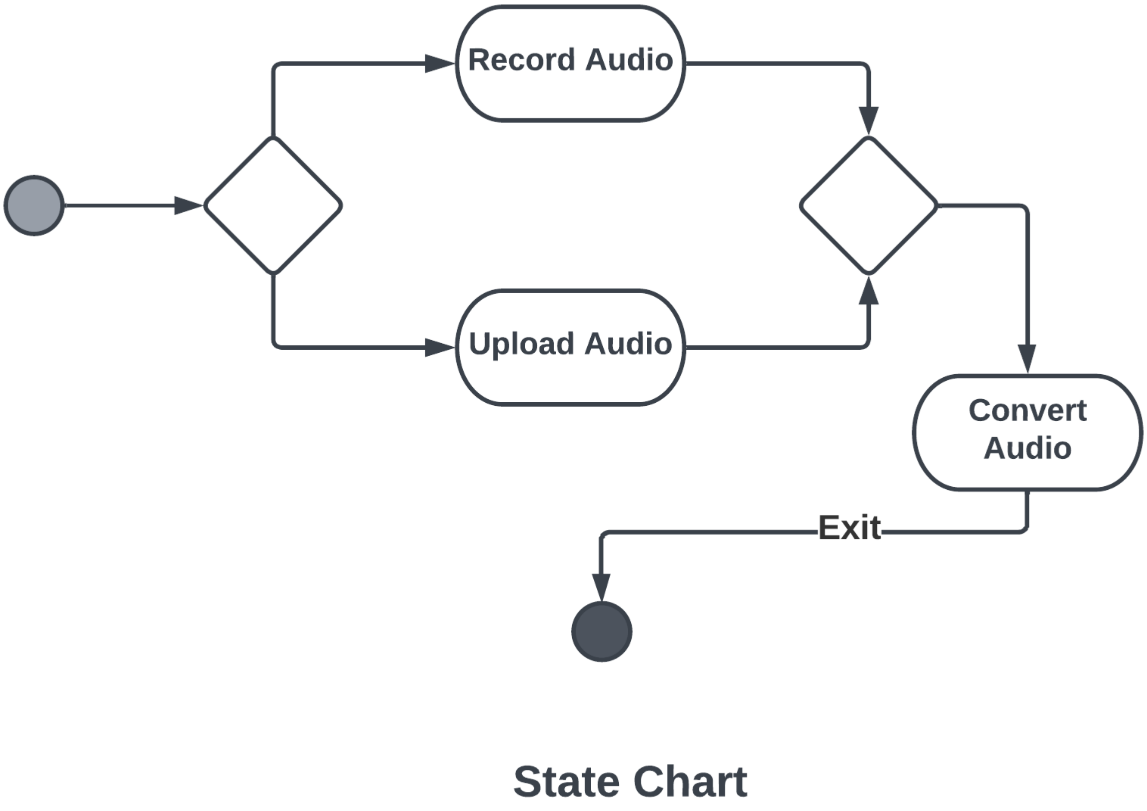


Figure 4.9: State chart Diagram

4.3.6 Component Diagram

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development. Figure shows component diagram for Speech to text Ahirani language.

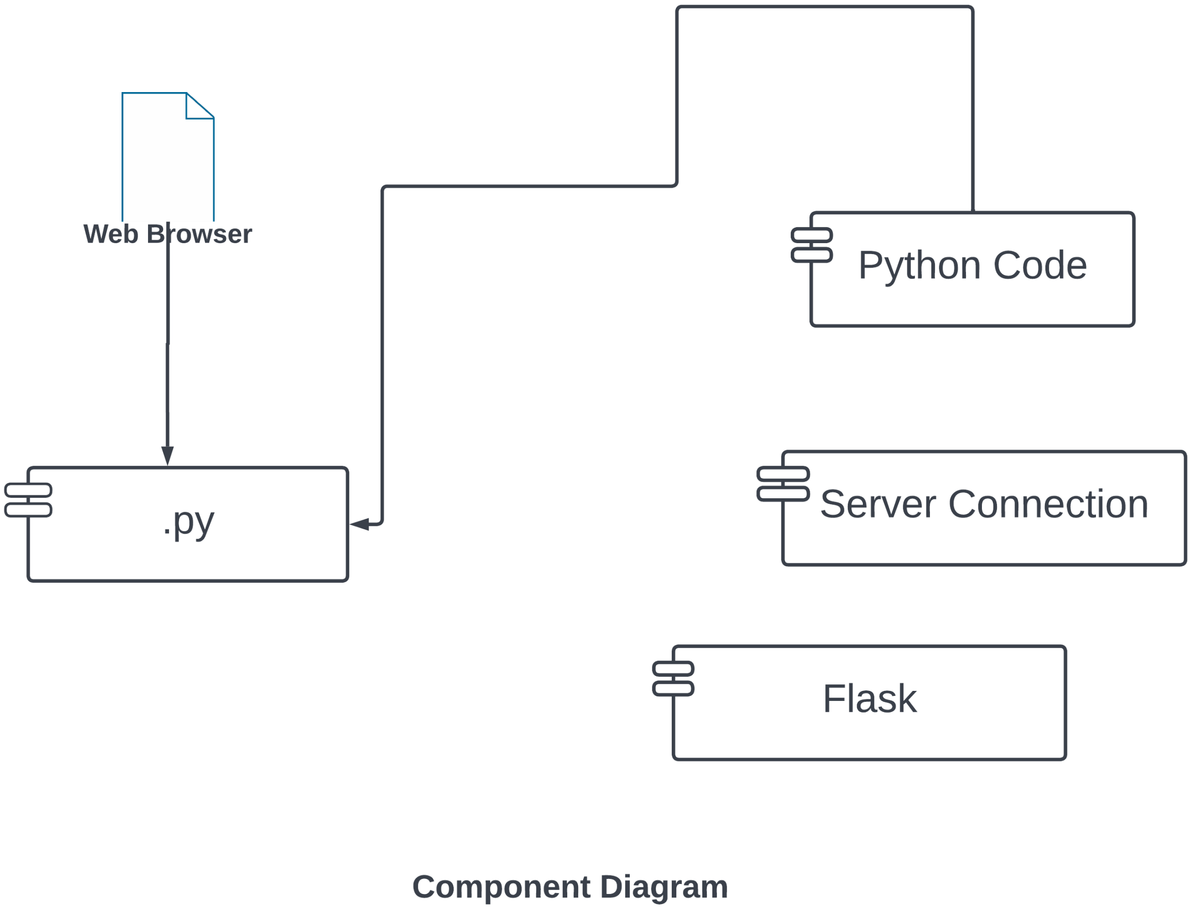


Figure 4.10: Component Diagram

4.3.7 Deployment Diagram

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware. Figure shows deployment diagram for Speech to text for Ahirani language.

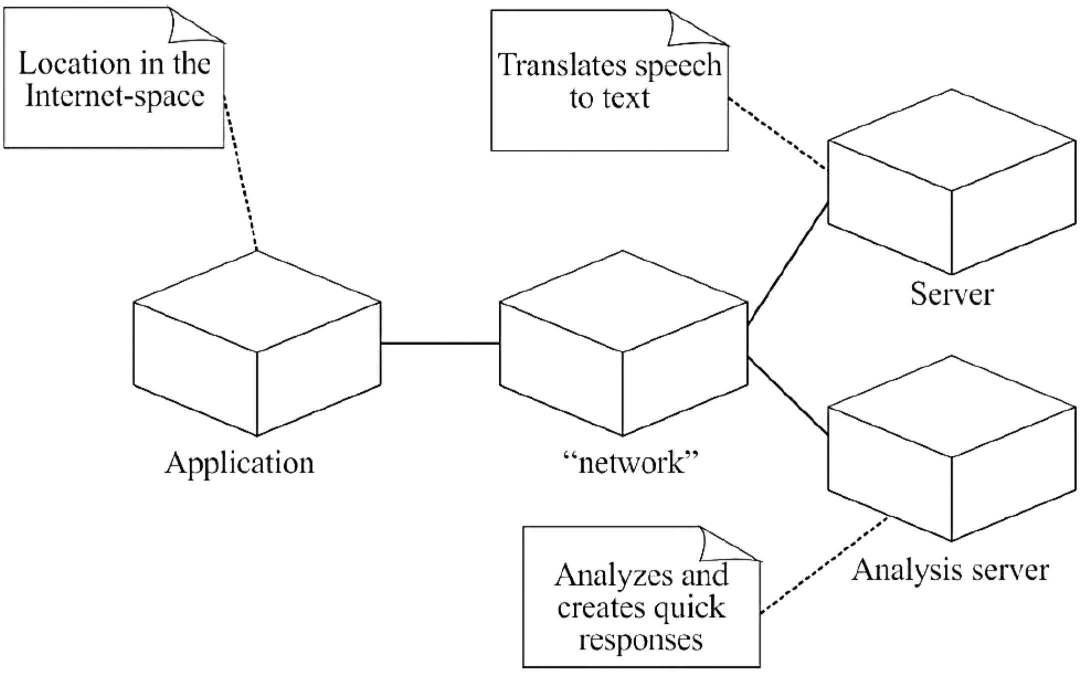

Location in the Internet-space
Translates speech to text
Server
Application
"network"
Analysis server
Analyzes and creates quick responses

Figure 4.11: Deployment Diagram

4.4 Summary

This chapter contains important aspects of Designing of Model like the System Architecture, Data Flow Diagrams (DFDs), UML diagrams which contains the Blueprints of the system. They highlight the architecture, ow and sequential execution of the events. As Designing is one of the most important process for building a project and which also helps in coding phase .UML diagrams are useful for understanding features of project. The more you know about the subject, the more new information you can find on it. More summary: If you know that an event is very likely, it is not surprising when it happens, that is, to give you a little detail that it actually happened.

From the above statement we can establish that the amount of information obtained is equal to the probability of an event occurring. We can also say that as Entropy increases the profitability of knowledge decreases. This is because Entropy means an event is possible.

Pruning:

The Decision Tree in our original example is quite simple, but it is not so when the data set is huge and there are more variables to take into consideration. This is where Pruning is required. Pruning means the removal of those branches from our decision-making tree that we feel are not contributing much to our decision-making process.

Chapter 5

Implementation

Implementation of the proposed system involves the environment in which the system is implemented and the overall system development. The overall development of the proposed system requires suitable environment and proper resources for its successful completion. Implementation of proposed system consists of various techniques and algorithms. Section 5.1 talks about the Algorithms used in model building. The implementation details is discussed in Section 5.2. Section 5.3 discusses about the software and hardware development in details and finally the overall summary is discussed in Section 5.4.

This chapter gives a full description of how the Ahirani language speech recognition system was developed. The goal of the project was to build a robust whole word recognizer. That means it should be able to generalise both from speaker specific properties and its training should be more than just instance based learning. In the HMM paradigm this is supposed to be the case, but the researcher intended to put this into practice. As the time scope was limited and to be able to focus on more specific issues than HMM in general, the Hidden Markov Model toolkit (HTK) was used. HTK is a toolkit for building Hidden Markov Models (HMMs).HMMs can be used to model any time series and the core of HTK is similarly general-purpose. However, HTK is primarily designed for building HMM-based speech processing tools, in particular recognisers. Secondly to reduce the difficulties of the task, a very limited language model was used. Future research can be directed to more extensive language models. In ASR systems acoustic information is sampled as a signal suitable for processing by computers and fed into a recognition process. The output of the system is a hypothesis transcription of the utterances.

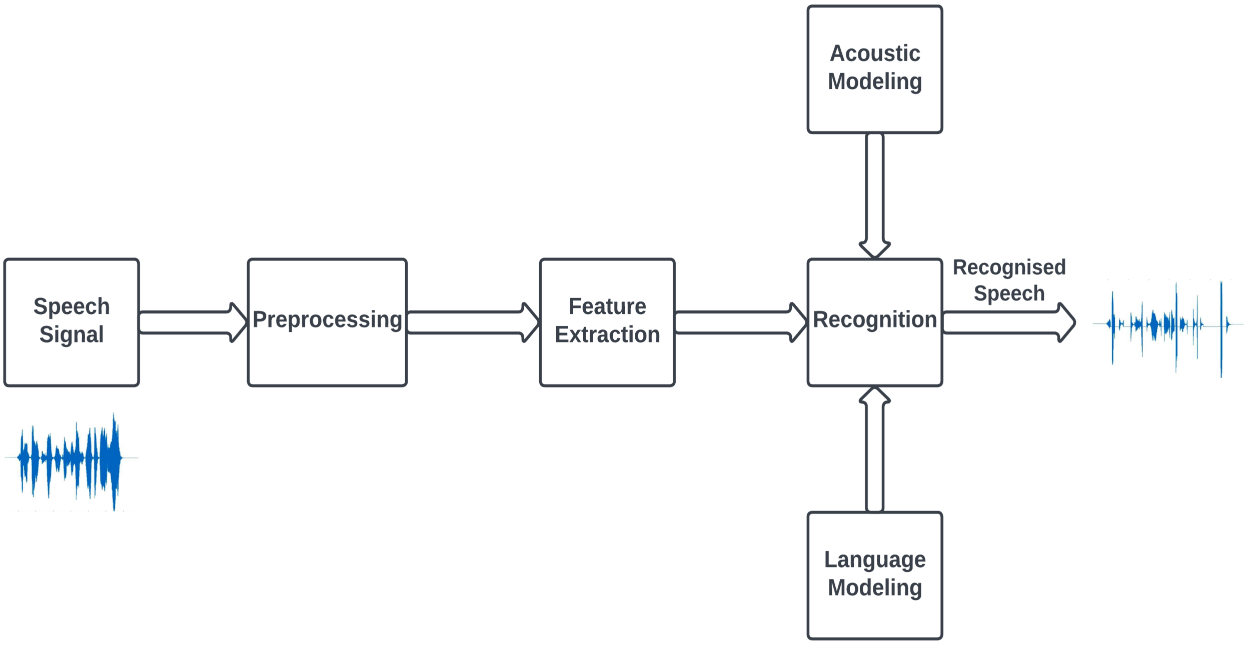


Figure 5.1: Components of an ASR system

There are a big number of different approaches for the implementation of an ASR but for this project the four major processing steps as suggested by HTK were considered namely data preparation, training, Recognition/testing and analysis.

5.1 Algorithm \Steps

Algorithm used for the proposed System is a Hidden Markovv Model (HMM). HMM is a statistical model that can be used to to recognize speech.

Steps involved are as follows:

- Creation of Speech Corpus
- Constructing a dictionary for the models
- Recording the data.
- Creating transcription files for training data
- Encoding the data (feature processing)
- (Re-) training the acoustic models
- Evaluating the recognisers against the test data
- Reporting recognition results

5.1.1 Workflow

Ahirani Speech recognition system is developed using HTK toolkit. Firstly, the HTK training tools are used to estimate the parameters of a set of HMMs using training utterances and their associated transcriptions. Secondly, unknown utterances are transcribed using the HTK recognition tools (Hidden Markov Model Toolkit, System is trained for 5 Ahirani sentences. Sentence model is used to recognize the speech.

- **Data Preparation:-**
  The first stage of any recogniser development project is data preparation. Speech data is needed both for training and for testing. In the system built here, all of this speech was recorded from scratch. Training and testing a speech recognition system needs a collection of utterances. System uses a data-set of 4 Sentences.The data is recorded using unidirectional microphones. Distance of approximately 5-10 cm is used between mouth of the speaker and microphone. Recording is carried out at room environment. Sounds are recorded at a sampling rate of 16000 Hz. Voices of four people (3 male and 1 female) are used to train the system. Each one is asked to utter each word Ten times. Thus giving a total of 320, (4*80) speech files. Speech files are stored in .wav format. The training data is used during the development of the system. Test data provides the reference transcriptions against which the recogniser's performance can be measured.

- **Feature Extraction:-**
  During this step, the data recorded is parameterized into a sequence of features. The technique used for parameterization of the data is Mel Frequency Cepstral Coefficient (MFCC). The very basic step for a speech recognition system is Feature Extraction of the speech signal. where the audio signal is filtered in terms of various parameters known as the feature extraction technique. The goal of feature extraction is to extract a set of properties or features of an utterance that have acoustic correlations to the speech signal, that is parameters that can somehow be computed or estimated through the processing of the signal waveform. Such parameters are termed features. It includes the process of measuring some important feature or properties of the signal such as energy, amplitude or frequency response (i.e. signal measurement), amplify these measurements with some meaningful derived measurements of signal (i.e. parameterize the signals), and statically conditioning and mapping these numbers to form observation vectors. For the audio signal analysis, we have used feature extraction technique known as Mel Frequency Cepstral Coefficient (MFCC).

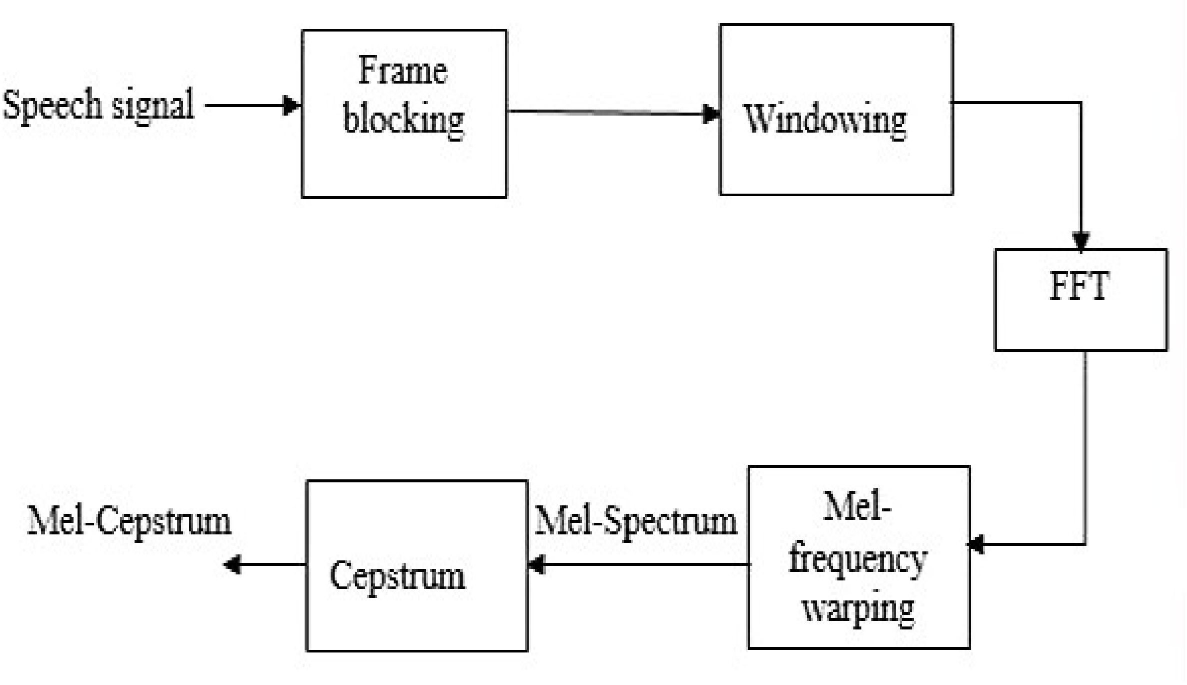


Figure 5.2: MFCC Extraction Process

To reduce noise in audio signal along with audio classification and identification of the speaker MFCC technique is used. The important parameter of speech signal in feature extraction method is Cepstral coefficients and pitch frequency, listed parameters are used for speech recognition, synthesis and verification of the speaker, etc. It summarizes the frequency distribution across the window size, to analyze both the frequency and time characteristics of the sound. The audio representation will allow to identify features for classification. In this extraction technique first, the speech is analyzed over short frame window. When the output of the FFT is passed through a Mel-filter then the Mel spectrum is obtained which performs the Cepstral coefficients.

- **Training The HMM:-**
  For training the HMM, a prototype model is created, which are then re-estimated using then data from the speech files. The prototype models are initialized using the HTK tool HInit which initializes the HMM model based on one of the speech recordings. Then HRest is used to re-estimate the parameters of the HMM model based on the other speech recordings in the training set.

- **Performance Evaluation:-**
  During evaluation, system is responsible for generating the transcription for an unknown utterance. The model generated during the training phase is responsible for evaluation. In order to evaluate the system performance, speakers are asked to utter

each sentence at least once a time. For testing four speakers are used. The recognition results are shown in table. Overall sentence-accuracy and sentence -error rate of the system is 82.8% and 17.2% respectively.

5.2 Software and Hardware for development in detail

The proposed system is developed using the Python, HTML , CSS and JS. The frontend end includes HTML, CSS, JS and Python is the back-end of the system and server used is the Flask (Localhost).

5.3 Implementation Environment

1. **Python:-**

   Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.NumPy, Pandas, matplotlib, scikit-learn are most popular libraries in python used for machine learning. NumPy is the fundamental package for scientific computing with Python. It mostly used for solving matrix problems. Pandas is the most popular machine learning library written in python, for data manipulation and analysis. Matplotlib, a great library for Data Visualization. Scikit-learn is one of the most popular ML libraries today. It supports most of ML algorithms, both supervised and unsupervised: linear and logistic regression, Decision tree, Random Forest, support vector machine (SVM), Naive Bayes classifier, gradient boosting, k-means clustering, KNN, and many others

5.4 Summary

In this chapter, the implementation details, implementation environment are described. In the chapter, System Testing is discussed.

Chapter 6

System Testing

Implementation of system testing is one of the important and difficult job and for testing the system, there are various test cases which are applied and results are considered. System testing is an empirical technical investigation conducted to provide stake holders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding errors. System Testing is a set of activities that can be planned in advance and conducted systematically. The proposed system is tested in parallel with the software that consists of its own phases of analysis.

Following are the various types of testing:-

- Unit Testing:- Unit testing is a software development process in which the smallest testable parts of an application, called units are individually and independently examined closely for proper operation. Unit testing can be done manually but is often automated. During unit testing some errors were raised and all of them were rectified and handled well. The result was quiet satisfactory and it worked well.

- Integration Testing:- Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. The system has been passed through the integration testing. The modules were combined and tested for their compatibility with other modules. The test was almost successful. All the modules coexisted very well, with almost no bugs.

- System Testing:- System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such,should require no knowledge of the inner design of the code or logic.

6.1 Black Box Testing

Black box testing is a method of software testing that examines the functionality of an application without peering into its internal structure or working. In black box testing tester is aware of what the software is suppose to do but is not aware of how it does it.

Black Box Testing is not a type of testing; it instead is a testing strategy, which does not need any knowledge of internal design or code etc. As the name "black box" suggests, no knowledge of internal logic or code structure is required. The types of testing under this strategy are totally based/focused on the testing for requirements and functionality of the work product/software application. The base of the Black box testing strategy lies in the selection of appropriate data as per functionality and testing it against the functional specifications in order to check for normal and abnormal behavior of the system.

- Functional Testing:- In this type of testing, the software is tested for the functional requirements. The tests are written in order to check if the application behaves as expected.

- Stress Testing:- The application is tested against heavy load such as complex numerical values, large number of inputs,large number of queries etc. which checks for the stress/load the applications can withstand.

- Load Testing:- The application is tested against heavy loads or inputs such as testing of web sites in order to find out at what point the web- site/application fails or at what point its performance degrades.

- Recovery Testing:- Recovery testing is basically done in order to check how fast and better the application can recover against any type of crash or hard- ware failure etc. Type or extent of recovery is specified in the requirement specifications.

- User Acceptance Testing:- In this type of testing, the software is handed over to the user in order to find out if the software meets the user expectations and works as it is expected to.

- Alpha Testing:- In this type of testing, the users are invited at the develop- ment center where they use the application and the developers note every particular input or action carried out by the user. Any type of abnormal behavior of the system is noted and rectified by the developers.

- Beta Testing:- In this type of testing, the software is distributed as a beta version to the users and users test the application at their sites. As the users explore the software, in case if any exception/defect occurs that is reported to the developers

6.2 White Box Testing

It is software testing technique in which internal structure, design and coding of software are tested to verify ow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, open box testing, transparent box testing, Code-based testing and Glass box testing.

White box testing is done to find out the following information as shown in below:-

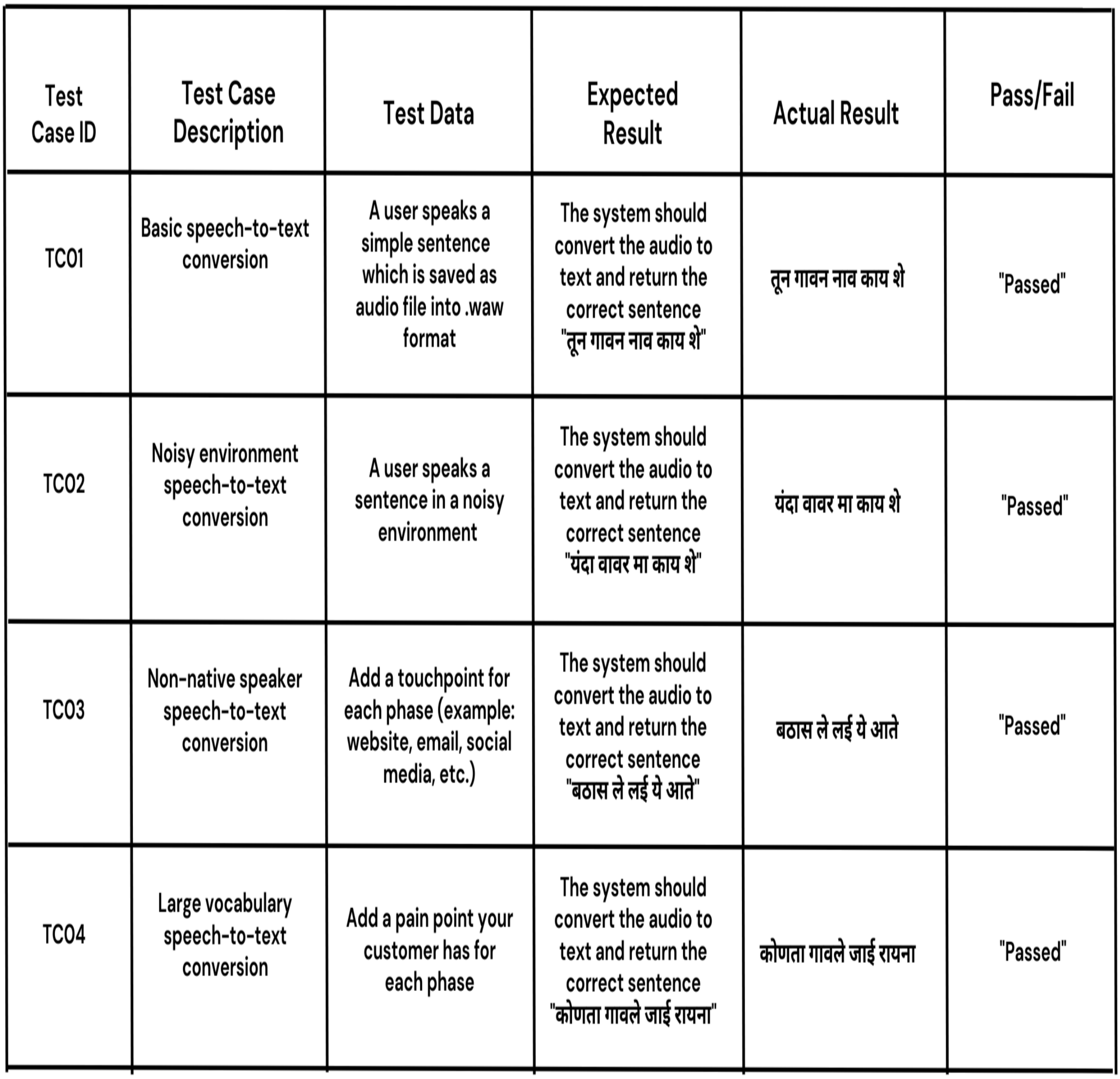
- Incorrect or missing functions
- To check flow of Input-output
- To improve design
- To check the usability
- Interface errors
- Errors or database access

White box testing strategy deals with the internal logic and structure of the code. White box testing is also called as glass, structural, open box or clear box testing. The tests written based on the white box testing strategy incorporate coverage of the code written, branches,paths, statements and internal logic of the code etc. In order to implement white box testing, the tester has to deal with the code and hence is needed to possess knowledge of coding and logic i.e. internal working of the code. White box test also needs the tester to look into the code and find out which unit/statement/chunk of the code is malfunctioning.

- Unit Testing:- The developer carries out unit testing in order to check if the particular module or unit of code is working fine. The Unit Testing comes at the very basic level as it is carried out as and when the unit of the code is developed or a particular functionality is built.
- Branch Coverage:- No software application can be written in a continuous mode of coding, at some point we need to branch out the code in order to perform a particular functionality. Branch coverage testing helps in validating of all the branches in the code and making sure that no branching leads to abnormal behavior of the application.
- Security Testing:- Security Testing is carried out in order to find out how well the system can protect itself from unauthorized access, hacking crack- ing, any code damage etc. which deals with the code of application. This type of testing needs sophisticated testing.

- Mutation Testing:- A kind of testing in which, the application is tested for the code that was modified after fixing a particular bug/defect. It also helps in finding out which code and which strategy of coding can help in developing the functionality eﬃectively.

6.3 Test Cases Identification and Execution



Test Case ID	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
TC01	Basic speech-to-text conversion	A user speaks a simple sentence which is saved as audio file into .waw format	The system should convert the audio to text and return the correct sentence "तून गावन नाव काय शे"	तून गावन नाव काय शे	"Passed"
TC02	Noisy environment speech-to-text conversion	A user speaks a sentence in a noisy environment	The system should convert the audio to text and return the correct sentence "यंदा वावर मा काय शे"	यंदा वावर मा काय शे	"Passed"
TC03	Non-native speaker speech-to-text conversion	Add a touchpoint for each phase (example: website, email, social media, etc.)	The system should convert the audio to text and return the correct sentence "बठास ले लई ये आते"	बठास ले लई ये आते	"Passed"
TC04	Large vocabulary speech-to-text conversion	Add a pain point your customer has for each phase	The system should convert the audio to text and return the correct sentence "कोणता गावले जाई रायना"	कोणता गावले जाई रायना	"Passed"

Figure 6.1: Test Cases Table

6.4 Summary

In this chapter, the system testing details, testing environment are described. In the next chapter, result and analysis are discussed.

Chapter 7

Results and Discussion

This chapter describes Discussion and the experimental results obtained for the proposed system.

7.1 Discussion

Speech to text is basically a process in which audio signals are converted into textual format based on the dataset provided. We have used the audio classification method with the help of a Hidden Markov Model to classify to classify the sentences into the appropriate category. The audio sample provided to the model is first filtered and then features from that audio are extracted with the help of a MFCC technique which extracts features from the audio and based on these features extracted audio is classified in its appropriate category. Once classified label assigned to it is provided as a output.

7.2 Results

Sample rate – Sample rates is basically number of audio samples recorded per unit second. 22050 is the default sample rate for reading the files using librosa. Depending on the libraries sample rate can vary..

2-D Array – Recorded samples of amplitude are shown by the 1st axis and number of channels are represented by the second axis respectively. The channels have two types– Monophonic and stereo. Monophonic has only one channel while the stereo has two channels.

Batch Size	**Epoch**	**Accuracy %**
110	38	46.8%
120	40	51.2%
140	50	64%
160	60	79.8%
180	80	82.8%

Table 7.1: Accuracy Benchmark Table

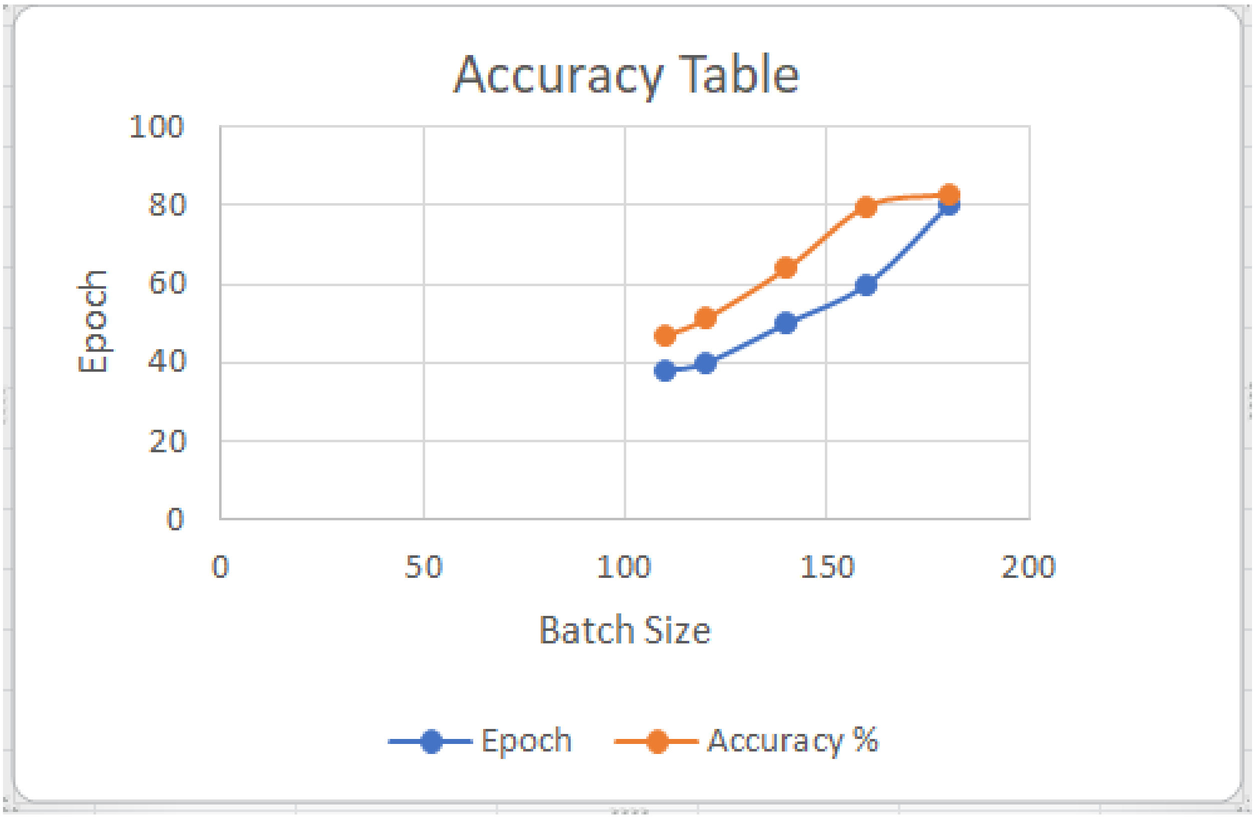


Figure 7.1: Linear relationship between Accuracy and Epoch

7.3 Summary

In this chapter, the result and analysis details, test cases and test results are described. In the next chapter, Conclusion and Future Scope are discussed.

Chapter 8

Conclusion And Future Scope

Conclusion

We presented a new Speech Corpus for Ahirani Language along with the large vocabulary of it along with that developed the speech recognition system for Ahirani language. where the system will recognize the short sentences using acoustic word model. The training of the system will be done using the Ahirani sentences. During the development of the system, the training data will be collected from the four different speakers, and accuracy achieved till date is 82.8%. The use of such an ASR solves the problem of technology acceptance in India, majorly in rural and remote parts by bringing human interaction closer to human-Machine interaction.

In future we have plan to create more robust speech corpus for the Ahirani language which will cover the wide variety of sentences and words and also to improve the accuracy of the system.

Bibliography

1. Ajay S. Patil, Automatic Speech Recognition for Ahirani Language Using Hidden Markov Model Toolkit (HTK) ,International Journal of Computer Science Trends and Technology (IJCST) – Volume 2 Issue 3, May-Jun 2014.

2. Uday Chandrakant Patkar, Dr. Suhas Haribhau Patil , Dr. Prasadu Peddi , Translation of English to Ahirani Language, International Research Journal of Engineering and Technology (IRJET) Volume 2 Issue 6,— June 2020.

3. Vinnarasu A., Deepa V. Jose, Speech to text conversion and summarization for effective understanding and documentation, International Journal of Electrical and Computer Engineering (IJECE) Vol. 9, No. 5, October 2019, pp. 3642 3648 ISSN: 2088-8708, DOI: 10.11591/ijece.v9i5.pp3642-3648.

4. Sunil S. Nimbhore, Ghanshyam D. Ramteke, Rakesh J. Ramteke, Implementation of English-Text to Marathi- Speech (ETMS) Synthesizer, IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278- 0661,p-ISSN: 2278-8727, Volume 17, Issue 1, Ver. VI (Jan – Feb. 2015), PP 34-43 www.iosrjournals.org.

5. P. P. Shrishrimal, R. R. Deshmukh, Vishal B. Waghmare, DEVELOP MENT OF ISOLATED WORDS SPEECH DATABASE OF MARATHI WORDS FOR AGRICULTURE PURPOSE, Asian Journal of Computer Science And Information Technology volume 2 Issue 7 July 2012

6. Dr. Rajeev B. Kharat, "A Model for Translation of English Words to Ahirani (Khandeshi) Words", International Journal of Latest Research in Humanities and Social Science (IJLRHSS) Volume 03 - Issue 02, 2020 www.ijlrhss.com —— PP. 46-48

7. Devaraja Adiga1, Rishabh Kumar, Amrith Krishna, Preethi Jyothi, Ganesh Ramakrishnan, Pawan Goyal, Automatic Speech Recognition in Sanskrit: A New Speech Corpus and Modelling Insights, India.

8. Gawali, Bharti W., Gaikwad, S., Yannawar P., Mehrotra Suresh C., Marathi Isolated Word Recognition System using MFCC and DTW Features (2010), Int. Conf. on Advances in Computer Science 2010, DOI: 02.ACS.2010. 01.73.

9. Kuldeep Kumar, R. K. Aggarwal, Hindi Speech Recognition System using HTK, International Journal of Computing and Business Research, Volume 2 Issue 2 May 2011.

10. Dr. Rajeev B. Kharat, A Model for Translation of English Words to Ahirani (Khandeshi) Words, International Journal of Latest Research in Humanities and Social Science (IJLRHSS), Volume 03 - Issue 02, 2020

11. Zue, V., Cole, R., Ward, W. (1996). Speech Recognition Survey of the State of the Art in Human Language Technology. Kauii, Hawaii, USA

12. Gupta, R, and Sivakumar G. "Speech Recognition for Hindi Language", 2006, M.Tech Project, IIT, Bombay,

13. Preeti Saini, Parneet Kaur, Mohit Dua, Hindi Automatic Speech Recognition Using HTK, International Journal of Engineering Trends and Technology (IJETT) – Volume 4 Issue 6, June 2013, pp. 2223-2229

14. Mohit Dua, R.K.Aggarwal, Virender Kadyan, Shelza Dua, Punjabi Automatic Speech Recognition Using HTK, International Journal of Computer Science Issues, Vol. 9, Issue 4, No 1, July 2012 p. 359-364

15. Aggarwal R.K., Dave Mayank, Implementing a Speech Recognition System Interface for Indian Languages, in proceedings of the IJCNLP-08 workshop on NLP for less priviledged Languages, pp. 105-112, Hyderabad, India, January 2008.

16. Paribesh Regmi, Arjun Dahal, and Basanta Joshi. Nepali speech recognition using rnnctc model. International Journal of Computer Applications, 178(31):1–6, Jul 2019.

Publications

1. Pooja Shirude, Mohit Chaudhari, Gaurav Baviskar, Mahesh kanhere , Towards understanding speech to text conversion for Ahirani language, International Journal of Computer Science Trends and Technology (IJCST) – Volume 11 Issue 2, Mar-Apr 2023 **Permanent URL:** http://www.ijcstjournal.org/volume-11/issue-2/IJCST-V11I2P2.pdf

2. Pooja Shirude, Mohit Chaudhari, Gaurav Baviskar, Mahesh kanhere , Towards understanding speech to text conversion for Ahirani language, IJSSBT- PRATIBHA Journal.

Participations

1. Secured first prize for presenting a paper titled ”Towards understanding speech to text conversion for Ahirani language” in Milestone 2K23 - National level event organized by SSBT'S College Of Engineering and Technology , Bambhori, Jalgaon, on 23 March 2023.

2. Presented a Poster on ”Towards understanding speech to text conversion for Ahirani language” in Milestone 2K23 - National level event organized by SSBT'S College Of Engineering and Technology , Bambhori, Jalgaon, on 23 March 2023.

3. Presented a poster on ”Towards understanding speech to text conversion for Ahirani language” in Avishkar 2022- A District level event organized by Kavayitri Bahinabai Chaudhari North Maharashtra university , Jalgaon, on 18 October 2022.