TAPI IN TELEPHONE QUALITY SPEECH DATABASE

MOHD SHUKRI BONGSU

A project report submitted in partial fulfillment
of the requirements for the award of the degree of
Master of Engineering
(Electrical – Electronics & Telecommunication)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

MAY 2007

To beloved wife and sons

ACKNOWLEDGEMENT

First of all, I would like to take this opportunity to thank my supervisor, Associate Professor Ir. Dr. Sheikh Hussein bin Shaikh Salleh. It is him who had made this project possible. He has shown me guidance, important advices, and inspiration throughout my project. Thank you, Sir, I will never forget your kindness and most of all, your dedication which I really felt.

Furthermore, to my beloved family, friends and fellow course mates must not be left out. I want to thank them for sharing and discussing knowledge with me and always give suggestions and opinion on my project, which I find very precious. I really appreciate the helping hands and encouragements given when I really need it.

Finally, I want to record my special thanks to Mr Amar for his full assistance and guidance along the way to successfully complete the project.

ABSTRACT

It is always been the speech recognition team's vision to be able to apply the developed speech technology in real world so that more people can benefit form it. One of the targets is towards telephony. People will be able to talk comfortably to computer through telephone to obtain certain information. Studies and effort have been carried out to improve the accuracy and efficiency of telephone speech recognition. This project aims to build a computer telephony using Telephony Application Programming Interface (TAPI) to collect telephone quality speech database, which are very useful in testing and improving certain speech recognition system's ability in recognizing telephone speech. First, discussion on TAPI itself will be presented. Then the whole system will be designed using TAPI. The implementation of the design will be done by using Visual Basic 6.0 as programming language. When the computer telephony is completed, speech samples will be collected to keeps as database. The speech samples will be collected through various type of Public Services Telephone Network. Parts of this database are the will be used for experiments to compare the performance of a certain speech recognition system that are trained in two ways: using soundcard quality speech (clean speech) as training samples and other one using telephone quality speech samples.

ABSTRAK

Ahli-ahli kumpulan teknologi pengecaman suara telah lama berhasrat untuk mengaplikasi system pengecaman suara pada system telefon. Justeru, usaha dan kajian telah dijalankan bagi meningkatkan prestasi suatu pengecaman suara supaya mempamerkan ketepatan yang lebih tingggi dan efisyen dalam mengecam suara melalui wacana telefon. Projek ini bertujuan untuk membangunkan sebuah sistem yang beroperasi sebagai telefoni komputer menggunakan TAPI. Fungsi sistem ini ialah untuk mengumpul pangkalan data suara berkualiti telefon. Pangkalan ini amat berguna dan penting untuk meningkatkan dan menguji kebolehan suatu sistem pengecaman suara. Implementasi rekabentuk tersebut dilakukan dengan menggunakan bahasa pengatucaraan Visual Basic 6.0. Setelah sistem selesai dibangunkan, kerja mengumpul pangkalan data dijalankan dengan menggunakan sistem tersebut. Kerja pengumpulan pangkalan data bagi sampel suara telefon melibatkan kepelbagaian rangkaian infa telefon yang sediada di Telekom Malaysia Bhd. Sebahagian daripada pangkalan data yang dikumpulkan akan digunakan dalam eksperimen bagi membandingkan prestasi suatu sistem pengecaman suara yang dilatih menggunakan suara terus. Manakala sebahagian daripadanya akan digunakan untuk melatih sistem pengecaman suara menggunakan kualiti suara telefon. Ini bertujuan untuk mendapatkan perbezaan prestasi kepada sesuatu sistem pengecaman suara yang dilatih menggunakan kualiti suara yang berbeza iaitu menggunakan kualiti suara terus dan juga suara telefon.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF GRAPHS

LIST OF ABREVIATIONS

ASR	-	Automatic Speech Recognition
PSTN	-	Public Services Telephone Network
TAPI	-	Telephone Application Programming Interface
WOSA	-	Windows Open Services Architecture
API	-	Application Programming Interface
PC	-	Personal Computer
FDM	-	Frequency Division Multiplexing
TDM	-	Time Division Multiplexing
DEL	-	Direct Exchange Line
MDF	-	Main Distribution Frame
CPE	-	Customer Premises Equipment
DP	-	Distribution Point
D' side	-	Distribution Side
RILL	-	Radio In Local Loop
PBX	-	Private Branch Exchange
ESS	-	Electronic Switch System
SPC	-	Stored Program Control
ROM	-	Read Only Memory
SDM	-	Space Division Multiplexing
PCM	-	Pulse Code Modulation
DNHR	-	Dynamic Non-Hierarchical Routing
INT	-	International
DTS	-	Digital Trunk Switch
DLS	-	Digital Local Switch
DRS	-	Digital Remote Switch

LPC	-	Linear Predictive Coding
MFCC	-	Mel-Frequency Cepstrum Coefficients
VQ	-	Vector Quantization
DTW	-	Dynamic Time Wrapping
HMM	-	Hidden Markov Model
DFT	-	Discrete Fourier Transform
LPCC	-	Linear Prediction Cepstrum Coefficients
ISDN	-	Integrated Service Digital Network
Tx	-	Transmission line

LIST OF APPENDICIES

CHAPTER 1

INTRODUCTION

1.1 Background of the problem

It is very common nowadays for people, especially university students to make a phone call to get some information like result or application status. Computer telephone usually answers the call automatically. All the caller need to do is just provide some information, for example identity card numbers by pressing telephone buttons and then wait for the response. The computer system at the other end of the line will search for desire information in database based on the callers' input.

This kind of system is indeed useful and convenient, as it can operate with consistent performance and longer time compared to a human telephony. However, the system can only handle inputs formed up by numbers or digits. It cannot handle alphabets, for instance, names. What shall we do if we want to know some information pertaining to persons name such as their extension telephone numbers and etc? The implementation of speech recognition technology through telephone should be best solution for this problem.

1.2 Project objective

The main objective of this project is to analyze the performance accuracy for Automatic Speech Recognition (ASR) Engine in recognizing Telephone Quality Speech for various type of PSTN Network Architecture comparing with Clear Quality Speech.

1.3 Scope of project

This project consists of 3 modules:

1. Design & Develop TAPI Application.
2. Database collection: Collective of speech samples through various type of Telephone Network.
   - I. Access network
     - o Copper Access Network
     - o Fiber Access Network.
   - II. Switching Network
     - o Connected to same Switching Network Element (Brand/Type).
     - o Connected to different Switching Network Element (Brand/Type).
   - III. Trunk or Junction Network
3. Performing evaluation of telephone speech quality using Automatic Speech Recognition (ASR) Engine.

CHAPTER 2

WHAT IS TAPI

2.1 Introduction

The telephony Application Interface (TAPI) is one of the most significant API sets to be released by Microsoft. The telephony API is a single set of function calls that allows programmers to manage and manipulate any type of communications link between the personal computer and the telephone line(s). This chapter providers a general overview of the Telephony API and how it fits into the WOSA (Windows Open Services Architecture) model.

2.2 The Telephony API Model

The TAPI design model is divided into two areas, each with its own set of API calls. Each API set focuses on what TAPI refers to as a “device”. The two TAPI devices are:

i. “Line devices” to model the physical telephony lines used to send and receive voice and data between locations.
ii. “Phone devices” to model the desktop handset used to place and receive calls.

2.3 Typical Configuration

The TAPI model is designed to function in several different physical configurations. There are four general physical configurations:

i. *Phone-based*
ii. *PC-based*
iii. *Shared* or *unified line*
iv. *Multiline*

2.3.1 Phone-Based Configuration

This configuration is best for voice-oriented call processing where the standard handset (or some variation) is used most frequently.

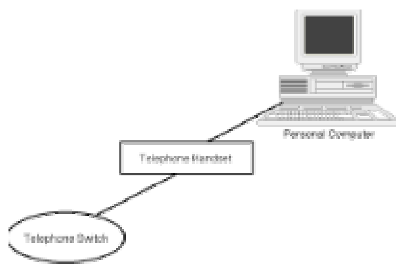

Figure 2.1: A typical phone-based TAPI configuration

This configuration is most useful when the telephone handset is the primary device for accessing the telephone line. Since the telephone rests between the PC and the switch, the PC may not be able to share in all the activity on the line.

2.3.2 Personal Computer Based Configuration

This configuration is best for data-oriented call processing where the PC is used most frequently for either voice or data processing.

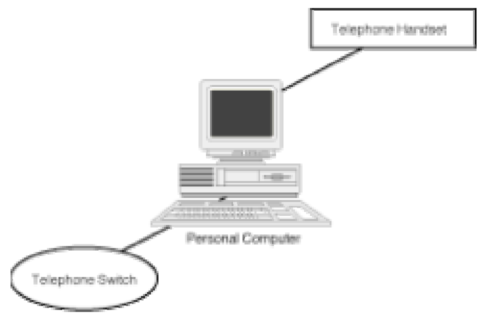

Figure 2.2: Typical PC based TAPI configuration

This configuration is most useful when the PC is the primary device for accessing the telephone line.

2.3.3 Shared or Unified Line Configuration

This is a compromise between phone-based and PC-based systems. It allows all devices to operate as equals along the service line.

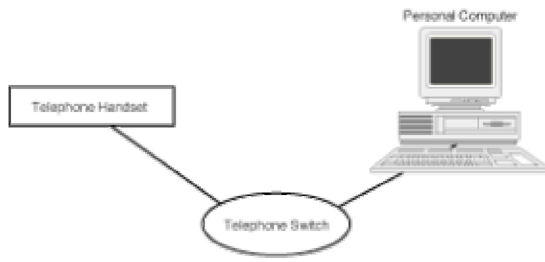

Figure 2.3: Typical shared line TAPI configuration

2.3.4 Multiline Configuration

The primary difference between this configuration and the others is that the PC acts as either a voice-server or a call switching center that connects the outside phone lines to one or more PCs and telephone handsets. The primary advantage of Multiline configurations is that we do not need a direct one-to-one relationship between phone lines and end devices (phones or PCs).

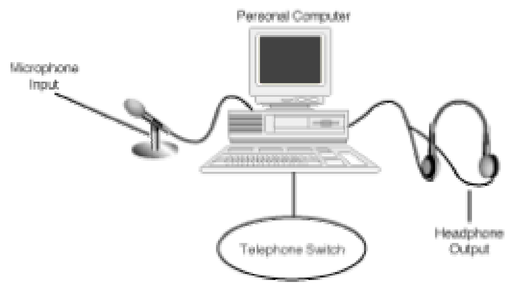

Figure 2.4: Typical unified line TAPI configuration

2.4 TAPI Architecture

The four different levels of TAPI services:

1. Assisted Telephony.
2. Basic Telephony
3. Supplemental
4. Extended Telephony

2.4.1 Assisted Telephony Services

The simplest form of TAPI service is Assisted Telephony. Under the Assisted Telephony interface, programmers can place outbound calls and check the current dialing location of the workstation.

2.4.2 Basic Telephony Services

Basic Telephony is the next level up in the TAPI service model. Basic Telephony function calls allow programmers to create applications that can provide basic in- and outbound voice and data calls over a single-line analog telephone.

2.4.3 Supplemental Telephony Services

The Supplemental Telephony functions provide advanced line device handling (conference, park, hold, forward, and so on). Access to these advanced services is dependent on the type of telephone line to which the workstation is connected. The Supplemental Telephony functions also allow programmers to handle service requests for multiple-line phones.

2.4.4 Extended Telephony Services

The last level of Telephony services is Extended Telephony. Extended Telephony service allows hardware vendors to define their own device-specific functions and services and still operate under the TAPI service model.

2.5 TAPI Hardware Consideration

The three primary types of telephony hardware for PCs:

- Basic data modems - can support Assisted Telephony services (outbound dialing) and usually are able to support only limited inbound call handling.
- Voice data Modems - capable of supporting the Basic Telephony services and many of the Supplemental services.
- Telephony cards - support all of the Basic Telephony and all of the Supplemental Telephony services, including phone device control.

2.6 A Quick Review of How Modems Work

Any information sent over the telephone line has to be in the form of sound waves. In order to accomplish this feat, hardware was invented to convert digital information into sound, and then back again from sound into digital information. Sending data over phones lines involves three main steps. First, a connection must be established between two modem devices over a telephone line. In the second step, the digital information is modulated into sound and then sent over the voice-grade telephone line to the modem. In the last step, the modem at the other end of the call converts (demodulates) the sound back into digital information and presents it to the computer for processing.