Gujarati Language Speech Recognition System for Identifying Smartphone Operation Commands

Jigisha K. Patel,Pritesh N. Patel,Paresh V. Virparia

Abstract— Natural Language Processing provides the facility of operating a system with speech and the system can interact with the user accordingly. Many speech recognition applications also provide the support for regional languages. In this paper, we would like to discuss the work that will add some facility to the state-of-the-art facility of speech recognition of Gujarati language. We have designed and developed a system that allows the users to give commands to their smartphone in Gujarati language for some basic facilities like calling, sending SMS, etc. The vocabulary includes total of 60 words consisting of Gujarati digits, some persons' name to be considered as a contact name and operational commands which yield the overall average recognition accuracy of 82.23%.

Keywords— Speech recognition, Phoneme, Hidden Markov Model (HMM), Java grammar, Lexeme

1. INTRODUCTION

Speech Recognition (SR) is the translation of spoken words into text which is popularly known as "Automatic Speech Recognition" (ASR), "Computer Speech Recognition", or "Speech to Text" (STT). The speech recognition component converts the user's speech into a sentence of distinct words, by matching acoustic signals against a library of phonemes—irreducible units of sound that make up a word. [20]. The goal of an ASR system is to accurately and efficiently convert a speech signal into a text message transcription of the spoken words independent of the speaker, environment or the device used to record the speech (i.e. the microphone). This process begins when a speaker decides what to say and actually speaks a sentence. The software then produces a speech wave form, which embodies the words of the sentence as well as the extraneous sounds and pauses in the spoken input. Next, the software attempts to decode the speech into the best estimate of the sentence. First it converts the speech signal into a sequence of vectors which are measured throughout the duration of the speech signal. Then, using a syntactic decoder it generates a valid sequence of representations. [21] Nowadays people follow tight schedules and timelines. They generally run out of time often. In this hectic life, people would like to save the time wherever it is possible. People like to have the easy to use gadgets for doing their various routine tasks. By keeping this in mind, we have developed an application that can serve the purpose. We have designed and developed a system that can detect the spoken commands for various basic operations of smartphone like calling, sending SMS, creating a group, searching a contact etc. Numbers of smartphone models are already available in the market

which provides the similar functionality through voice recognition feature but none of the smartphone operating system supports the Gujarati language speech recognition.

Hence, the biggest advantage of the application is that it has been designed to identify the Gujarati language. The system will provide the facility to the users to operate the mobile phone while doing the other tasks like driving or reading. It will be also helpful to old age people who may have difficulty in operating smartphones due to weak eyesight or some other problem. Moreover, the system has been developed in java technology which is open source that makes the system easy to be adopted after little modifications by various mobile operating systems that supports java for embedding the Gujarati speech recognition functionality to their devices.

2. INTRODUCTION TO SPHINX4

CMUSphinx toolkit is a leading speech recognition toolkit with various tools used to build speech applications. CMUSphinx toolkit has a number of packages for different tasks and applications [2]. Sphinx-4 is one of the packages provided by CMUSphinx, which is a state-of-the-art speech recognition system written entirely in the Java programming language. It was created via a joint collaboration between the Sphinx group at Carnegie Mellon University, Sun Microsystems Laboratories, Mitsubishi Electric Research Labs (MERL), and Hewlett Packard (HP), with contributions from the University of California at Santa Cruz (UCSC) and the Massachusetts Institute of Technology (MIT). [5]

Sphinx-4 is an HMM-based speech recognizer. HMM stands for Hidden Markov Models, which is a type of statistical model. In HMM-based speech recognizers, each unit of sound (usually called a phoneme) is represented by a statistical model that stands for the distribution of all the evidence (data) for that phoneme. This is called the acoustic model for that phoneme [10]. The Figure 1 shows the architecture diagram of Sphinx 4 tool kit.

When the recognizer starts up, it constructs the front end (which generates features from speech), the decoder, and the linguist (which generates the search graph) according to the configuration specified by the user. These components will in turn construct their own subcomponents. For example, the linguist will construct the acoustic model, the dictionary, and the language model. It will use the knowledge from these three components to construct a search graph that is appropriate for the task. [10]

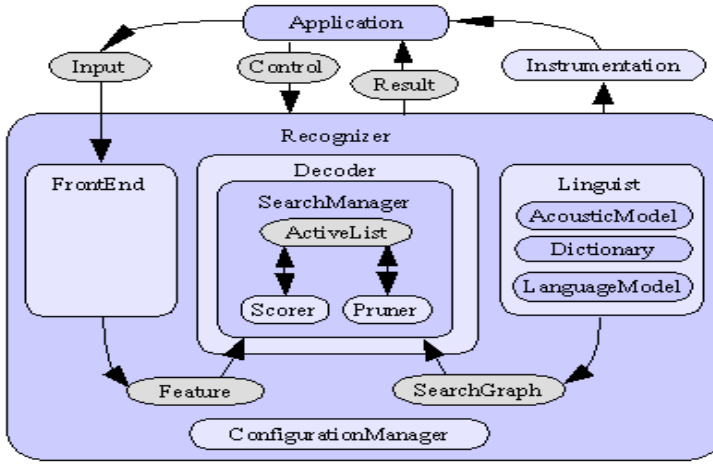

Fig.1 Block diagram representing the architecture of Sphinx4 [10]

Most of these components represent interfaces. There can be different implementations of these interfaces. For example, there are two different implementations of the search manager. Then, how does the system know which implementation to use? It is specified by the user via the configuration file, an XML-based file that is loaded by the configuration manager. In this configuration file, the user can also specify the properties of the implementations. [10]

Sphinx-4 currently implements a token-passing algorithm. Each time the search arrives at the next state in the graph, a token is created. A token points to the previous token, as well as the next state. The active list keeps track of all the current active paths through the search graph by storing the last token of each path. A token has the score of the path at that particular point in the search. To perform pruning, we simply prune the tokens in the active list. [10] When the application asks the recognizer to perform recognition, the search manager will ask the scorer to score each token in the active list against the next feature vector obtained from the front end. This gives a new score for each of the active paths. The pruner will then prune the tokens (i.e., active paths) using certain heuristics. Each surviving paths will then be expanded to the next states, where a new token will be created for each next state. The process repeats itself until no more feature vectors can be obtained from the front end for scoring. This usually means that there is no more input speech data. At that point, we look at all paths that have reached the final exit state, and return the highest scoring path as the result to the application. [10]

3. THE SYSTEM

We have developed a limited vocabulary speaker independent system that can recognize the basic operational commands for operations of smartphone devices in Gujarati language. Very limited number of commands has been considered as the input for the system. For various commands of mobile operations, equivalent Gujarati language word has been considered. The vocabulary includes total of 60 words consisting of Gujarati digits, some persons' name to be considered as a contact name and operational commands as listed in Table-1 with their meaning.

The Sphinx4 toolkit have been used as a recognizer component but that has to be modified as originally it is designed to recognize English language words; so that the pronunciation of Gujarati language words can be identified accurately. Hence, the grammar and phoneme sequences have been created for the system and have been embedded to the toolkit.

Table 1: Vocabulary in Gujarati with meaning

	Lexeme	In Gujarati	Meaning
	Digits		
1	Ek	Aok	Digit 1
2	Be	bao	Digit 2
3	Tran	~aNa	Digit 3
4	Char	caar	Digit 4
5	Panch	paaMca	Digit 5
6	Chha	C	Digit 6
7	Saat	saata	Digit 7
8	Aath	AaZ	Digit 8
9	Nav	nava	Digit 9
	Contact Persons' Name		
10	Kishan	ikrIT	--
11	Anuraag	Anauraga	--
12	Jigishaa	iJgaISaa	--
13	Moksh	maaoxa	--
14	Chintan	icaMtana	--
15	Nayanaa	nayanaa	--
16	Meetaa	maItaa	--
17	Shiv	iSava	--
18	Kirit	ikrIT	--
19	Ami	AmaI	--
	Operational Commands		
20	Aagad	AagaLa	Next/Forward
21	Bahaar	bahar	Out
22	Banaavo	banaavaao	Create
23	Bandh	baMQa	Close
24	Bataavo	bataavaao	Show
25	Call	kaola	Call

26	Chaalu	caalau	Start
27	Door	dUr	Delete
28	Google	gaugala	Google
29	Joddo	JaoDao	Combine/Paste
30	Juth	JUqa	Group
31	Kaapo	kapaao	Cut
32	Kaapi	kapaI	Cut
33	Karo	krao	Do
34	Kholo	Kaaolaao	Open
35	Lagaavo	lagaavaao	Dial
36	Lakho	laKaao	Type
37	Moklo	maaoklaao	Send
38	Nakal	nakla	Copy
39	Paachad	paaCLa	Previous/Back
40	Pasand	pasaMd	Select
41	Phone	Faona	Call/Dial
42	Prasaaran	pa`saarNa	Broadcast
43	Pratiko	pa`itakao	Symbols
44	Prayojak	pa`yaaoJk	Menu
45	Rad	rd	Discard/Delete
46	Saachvo	saacavaao	Save
47	Saaf	saaf	Clear
48	Sampark	saMpak-	Contact
49	Samparko	saMpak-ao	Contacts
50	Sandesh	saMdoSa	Message
51	Sandeshaa	saMdoSaa	Messages
52	Sandesho	saMdoSaao	Message
53	Sangeet	saMgaIta	Music
54	Screen	sk`Ina	Screen
55	Sharu	Saru	Start
56	Shodho	SaaoQaao	Search
57	Sopo	saaoMpaao	Submit
58	Umero	]maorao	Add
59	Vaatcheet	vaatacaIta	Chat
60	Vikalp	ivaklpa	Options

Table-2 shows the possible list of commands that can be used for the smartphone operations.

Table 2: List of commands

Command in Gujarati	Meaning
Faona krao	Dial call
Faona lagaavaao	Dial call
ivaklpa bataavaao	Show options
saMdoSa saacavaao	Save Message
saMdoSa banaavaao	Create Message

saMdoSa rd krao	Delete Message
saMdoSa saaf krao	Clear Message
saMdoSa maaoklaao	Send Message
saMdoSa pa`saarNa krao	Broadcast Message
saMdoSa bataavaao	Show Message
saMdoSa nakla krao	Copy Message
saMdoSa JaoDao	Paste Message
saMgaIta Saru krao	Play Music / Start Music
saMgaIta baMQa krao	Stop Music
gaugala Saru krao	Start Google
pa`yaaoJk bataavaao	Show Menu
pa`itakao]maorao	Add Symbols / Smileys
saMpak- banaavaao	Create Contact
saMpaka-o SaaoQaao	Search Contacts

The list of commands shown in table-2 is only the commands used for testing of the system with majority of the commands for SMS operations. More commands can be generated with the combination of various words used in vocabulary. For an instance, in music player, commands like next song and previous song can be generated with the combination of words like paaCLa (next) and AagaLa (previous). We have not covered the operational commands for dialing a call into much detail as it is already have been considered in some of our previous research work publications [5].

Java Speech Grammar Format (JSGF) has been used to create grammar used in the system. The Java Speech Grammar Format (JSGF) is a BNF-style, platform-independent, and vendor-independent textual representation of grammars for use in speech recognition [18]. Grammars are used by speech recognizers to determine what the recognizer should listen for, and so describe the utterances a user may say [19]. The grammar used in the system is not augmented as we have used the dictation grammar functionality. Hence the end of the command is considered by the longer pause in the speech input. The performance of the recognition greatly depends on the phoneme sequences that are used in the system. Most of the recognizers are developed originally for recognizing the English language which will not allow the recognizer to efficiently identify the speech of some other language. To overcome this situation, the phoneme sequences used in the recognizer toolkit have been studied and after the analysis we have generated our own set of phoneme sequences for all the words that we have incorporated in vocabulary. To provide the higher accuracy rate, we have used more than one sequence for a many of the words in our vocabulary by considering that application is speaker independent and different speakers may pronounce several words differently. Once the grammar has been created and the phoneme sequences have been generated according to the requirement of

the application, these changes have to be added to the recognizer toolkit. After the required modification the recognizer can recognize the suggested list of command as shown in Table-2. Once the speech command is converted into a text command, it can be executed by the mobile operating system. The system has been developed on the java platform giving it the benefits of platform independence, portability and adaptability. Hence the system can be converted into the web service which can be added to the mobile operating system as a feature or as add-in functionality.

4. RESULTS & DISCUSSIONS

During the testing of the system, we found difficulty in recognition of words like pa`yaaoJk, pa`itakao, saMpak- and pa`saarNa. These four word are having the adjacent consonants p and r in Gujarati language. Hence, we have tried various phoneme sequences of such words and finally we have done several modifications in the primary phoneme sequences. After the required modifications, the application has been tested in the laboratory environment using the head-mounted microphone by 20 speakers including 10 females and 10 males having the age between 22 and 36. The speaker was allowed to pronounce the word however he/she likes. We have obtained 82.23% overall average recognition accuracy rate of the system with the minimum accuracy of 65.17% and maximum accuracy of 94.14% among all the 20 speakers. The recognition accuracy rate for the female speakers is 78.34% and the same for the male speakers is 87.54%. The reason for this gap between the female and male recognition accuracy rate may be effect of vocal strength that comes with the gender features which also can be considered as the one of the future enhancement feature. We have not analysed the system for the age criterion as we were not able to get adequate range of age among the speakers.

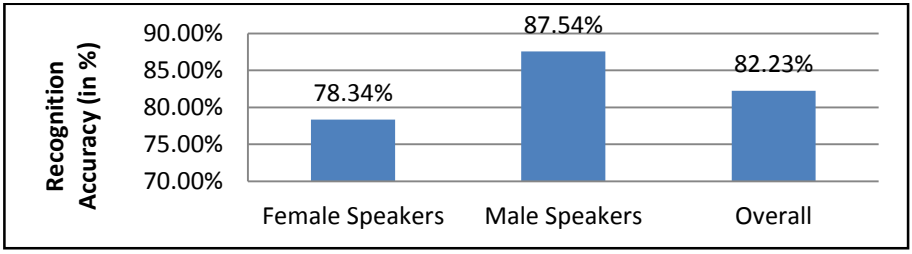

Figure 2: Recognition Accuracy

The system is able to handle majority of the variations in the pronunciations by different speakers. As the whole experiment has been carried out in laboratory environment, the performance may slightly decline in noisy surrounding.

5. CONCLUSION

We conclude that we have developed and tested the speaker independent limited vocabulary speech recognition system that is capable to recognize the operational commands in Gujarati language speech for the smartphone operations with average recognition accuracy rate of 82.23%. The system has been tested on 10 male and 10 female speakers in the laboratory environment. If the noise filtering capabilities will be added into the system then the performance of the system may increase.

6. REFERENCES

[1] Jurafsky, Martin – "Speech and Language Processing", Pearson, 2000
[2] Gunnar Fant - "Speech Acoustics and Phonetics: Selected Writings" [e-book], Springer, 2006
[3] Hinrich Schtze - "Foundations of statistical natural language processing" [e-book], MIT Press,1999
[4] Ms. Jigisha Patel, Mr. Pritesh N. Patel, Dr. P. V. Virparia – "Acoustic and Phonetic Confusions in Accented Gujarati Speech Recognition" : National Journal Of Engineering Science And Management (ISSN 2249 -0264) Bhopal
[5] Ms. Jigisha Patel, Mr. Pritesh N. Patel, Dr. P. V. Virparia – "Voice Enabled Telephony Commands Using Gujarati Speech Recognition" : International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X , Volume 3, Issue 10, October 2013, [Impact Factor: 2.080, Indexed]
[6] Ms. Jigisha Patel, Mr. Pritesh N. Patel - "Dialectical issues in speech recognition for Gujarati language", in the Proceedings of the National Conference on Advances in Computing-2011, North Maharashtra University
[7] http://cmusphinx.sourceforge.net/sphinx4/#what_is_sphinx4
[8] http://cmusphinx.sourceforge.net/wiki/tutorialoverview
[9] http://www.speech.cs.cmu.edu/tools/lmtool-new.html
[10] http://cmusphinx.sourceforge.net/sphinx4/#architecture_and_api1
[11] http://cmusphinx.sourceforge.net/sphinx4/#download_and_install
[12] https://javacc.java.net/doc/javaccgrm.html
[13] http://cmusphinx.sourceforge.net/wiki/sphinx4:jsgfsupport
[14] http://cmusphinx.sourceforge.net/sphinx4/javadoc/edu/cmu/sphinx/jsgf/JSGFGrammar.html
[15] http://www.w3.org/TR/speech-grammar/
[16] http://docs.oracle.com/cd/E17802_01/products/products/java-media/speech/forDevelopers/jsapi-doc/javax/speech/recognition/RuleGrammar.html
[17] http://www.ling.helsinki.fi/kit/2004s/ctl310gen/L7-Speech/JSAPI/Recognition.html
[18] http://cmusphinx.sourceforge.net/doc/sphinx4/edu/cmu/sphinx/jsgf/JSGFGrammar.html
[19] http://www.w3.org/TR/jsgf/
[20] http://oxygen.lcs.mit.edu/Speech.html
[21] http://support.docsoft.com/help/whitepaper-asr.pdf

7. AUTHORS' PROFILE

Ms. Jigisha K. Patel is working as a Lecturer at Dept. of Computer Science, Sardar Patel University. Her area of interests is Natural Language Processing. She has more than 7 years of experience in academic and research. She is pursuing Ph.d. from Sardar Patel University in the area of Speech Recognition. She has published 6 papers in international and national journals. She has also attended more than 5 conferences, seminars, workshops etc.

Mr. Pritesh N. Patel is a Research Scholar at Dept. of Computer Science, Sardar Patel University. His area of interests is Mobile computing and Natural Language Processing. He has more than 7 years of experience in academic and industry as well. He is pursuing Ph.d. from Sardar Patel University in the area of Mobile Computing. He has published 6 papers in international and national journals. He has also attended more than 5 conferences, seminars, workshops etc.

Prof. Paresh V. Virparia is working as a Director and Professor at Dept. of Computer Science, Sardar Patel University. His area of interests is Computer Simulation & Modeling, Data Mining, Networking and IT enabled services. He has more than 25 years of experience in academic and research. He is a Ph.d. from Sardar Patel University. He has guided 7 research scholars and 7 are currently under his guidance. He has published more than 40 papers in international journals and 16 papers in national journal. He has also attended more than 40 conferences, seminars, workshops etc.