International Journal of Latest Trends in Engineering and Technology
Vol.(11) Issue(4), pp.045-052
DOI: http://dx.doi.org/10.21172/1.114.09
e-ISSN:2278-621X

EVALUATING FACTORS FOR DEVELOPING OPEN SPEECH API FOR TYPICAL PUNJABI LANGUAGE & COMPARATIVE STUDY

Er. Karamjot Kaur[1], Dr. Pardeep Singh Cheema[2]

Abstract- The paper is going to evaluate the development factors of best automatic speech recognition API's to this effect in the development of cloud based open speech API in context to rise the utilization of undiluted words of Punjabi language. There are approximately 7000 currently spoken languages in the world. There are total 1635 Indian languages. 53 of Indian languages are put up as endangered languages and 197 are between endangered and vulnerable languages. The minority mother tongues are dying slowly. Punjabi is 10th most widely spoken language in the world & 5th most spoken native language in Canada but continuously losing its precious words due to dilution with other languages. Punjabi is written with Gurmukhi script; it's a meaning changing language with accent and lexical tone. Most of the history, literature and great holy scripture of the Sikh religion is written in Punjabi language. In the today's world if internet there are many speech recognition systems are available for most of languages and especially for English language but for Punjabi languages a very little work is done, which is limited to published work only without real life implementation. The existing speech recognition API's are not supporting the Punjabi language. To recover from the current issue and regenerating the undiluted words, the only solution is the development of an open and self-enriching speech recognition API in which users will contribute to train it into their own necessity and effortlessly integrate with the own applications, software's and websites.
Keywords – Open Speech API, Punjabi Speech API, Self-enriching Speech API, Speak Punjabi.

1. INTRODUCTION

Automatic Speech Recognition (ASR) is the way of conversion of speech taken through microphone into text (vice-versa) in real time or to perform particular task through speech interface. The main challenge is the response time and accuracy. The factors which affects the accuracy of automatic speech recognition systems includes system factors i.e. system is speaker-dependent or speaker-independent, environmental factors: recognition is done in noisy or noisy free environment, speaker factors: accent, age and gender of speaker, input factors: which type of microphone is used i.e. wired or wireless and recorded speech is used for recognition or live speech is used. The paper includes the automatic speech recognition system classifications, best selected speech recognition tools like Google Docs Voice Typing: the product of Google, Microsoft's Windows Speech Recognition and Nuance's Dragon Naturally Speaking. Paper focused on the development tools, programming language, simulator, speech database, architecture, techniques and algorithms, platform dependability and usability of these tools. A comparative evaluation of these tools has been made based on their development process.

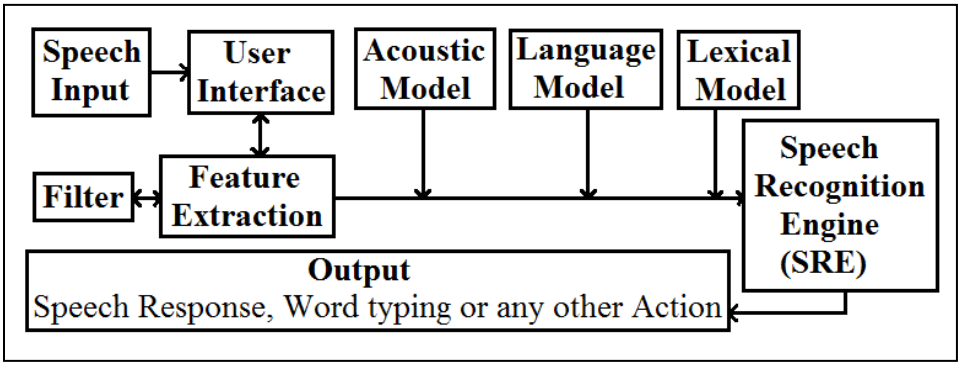

Figure 1: Basic Architecture of Automatic Speech Recognition

This section covers the advantages and classifications of automatic speech recognition systems (ASR), basic terms used in ASR systems, the purpose of this research work, reasons of language's dying and issues with existing speech API's. The rest

[1] Research Scholar, Department of Computer Science, Eternal University, Baru Sahib, HP, India
[2] Professor, Department of Computer Science, Eternal University, Baru Sahib, HP, India

of the paper is organized as follow: Section II describes the literature review, Section III evaluates the development factors of existing speech API's in context with architectural comparison between Google docs voice typing (using Google Speech API), Microsoft's windows speech recognition (using Microsoft Speech API) and Nuance's dragon naturally speaking (using Dragon Speech API), Section IV tells about the proposed model, proposed likely results and benefits over the existing models, Section V discuss two experiments performed in the MATLAB for the plotting of a live audio signal (recorded for 60 seconds) and a previously recorded audio signal. These experiments were performed in order to familiar with MATLAB because Google test the two modules of noise reduction in MATLAB using Wiener filter. The remaining sections discuss about the conclusion and the future scope.

1.1 Advantages of Speech Recognition System –

Speech recognition feature converts the user interface into better with simpler communication, creates a curious and user-friendly environment. System becomes flexible for the physically disabled users. Its feel something magicianly and users were encouraged to use the system. It reduces the typing time for large documents into half. Play a nice role in the student learning system where students can easily learn new things and languages with practices the pronunciation and communication skills. With cloud, features can be delivered to users with platform independence, reduced cost, reduced resources, better recovery, portability, openness etc.

1.2 Basic Terms used in Automatic Speech Recognition –

Accuracy: The Word Error Rate is used to measure the accuracy of speech recognizer. It can be defined as, how much the speech recognizer can correctly recognize the user speech input.

Speed (Response Time): The speed of automatic speech recognizers measures with the real-time factors which are clearly defined in abstract i.e.

- System is speaker-dependent or speaker-independent.
- Noisy or noisy free environmental.
- Accent age and gender of speaker.
- Type of microphone.
- Speech is real or live.

The speech recognition engine (SRE): Recognize the input speech by matching it with the grammar.

The speech synthesis engine: Simulates the human speech i.e. convert the text to speech for giving response to the user.

Speech recognition API: Provide the interface for speech recognition and speech synthesis.

Filter: Filter the input utterance form the noise.

Phonemes: Abstract representations of the smallest and semantic unit of speech signal.

Feature extraction: The original speech signal is portioned into fixed-sized frames known as acoustic frames.

Lexical model: Expound the words of vocabulary and send these words (lexicon units) to language model to generate acoustic model.

Language model: Takes the lexicon units from lexicon model & predict the input speech. It puts the lexicon units into a sequence for pronunciation.

Acoustic model: Statistical representation of the input speech which describes the behavior of the lexicons.

Speech Database: Stores all speech data for reorganization of user input, giving response to the user and perform particular operation according to the user input.

Word Error Rate (WER): It finds the misrecognition occurs during the speech recognition process through recognizer by matching the input speech with grammar, vocabulary and speech in language model.[1] It is calculated by the following formula:

$$\text{WER} = \frac{\text{Number of Substitution} + \text{Insertions} + \text{Deletions}}{\text{Total Number of Words}} \qquad (1)$$

Semantic Quality (Web Score): Some grammatical errors do not change the results so as the name suggests it finds the total number of correct results from the total number of speech input given by the user.

$$\text{Web Score} = \frac{\text{Number of Correct Search Results}}{\text{Total Number of Speech queries}} \qquad (2)$$

Perplexity (PPL): It defines the quality of the language model. The language model can easily predict, what the user can say next if the PPL is low.

Out-of-Vocabulary Rate (OOV): It defines the percentage of spoken words by the user which are not in the language model vocabulary.

Latency: It can be defined as the delay time in seconds between the user speech input and response from the recognition system.

1.3 Classifications of Automatic Speech Recognition System –

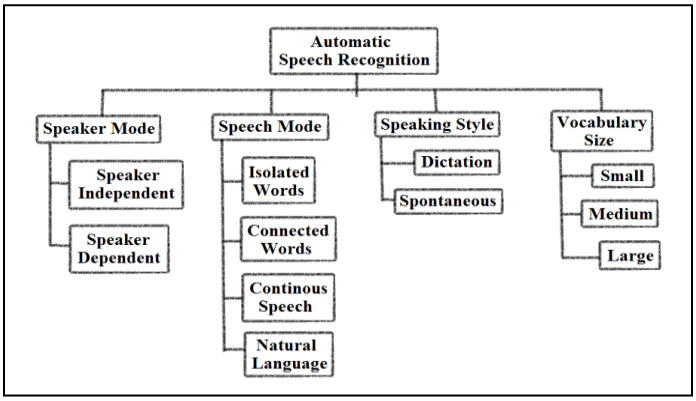

Figure 2:Classifications of Automatic Speech Recognition[2]

An ASR system can be classified according to the speaker mode, speech mode, speaking style and vocabulary size. According to speech mode the system can be classified into speaker dependent and speaker independent systems. Speaker dependent systems are dependent on the speaker to recognize and require that the system must be trained by the speaker before the recognizer starts recognition, but speaker independent systems can be used by any speaker without train the system. According to the Speech Mode the system can be classified as isolated words system, connected words system, continuous speech system and natural language systems. Isolated word systems recognize only clearly spoken one-word speech input at a time. Connected word systems recognize can recognize several words together with the requirement that words should be clearly spoken and must give pause between words to separate them so that the system can identify that from where the word starts and where ends. Continuous speech systems able to recognize the continuous speech signal without the requirement of pauses but the words must be clearly spoken. Natural language systems can recognize the natural form of speech input as the humans normally speak in the daily life. According to the speaking style the system classifications are based on the speaking style of the speaker i.e., the speaker uses the dictation style like a news caster or spontaneous style like naturally speaking. The vocabulary size of the system is the most important aspect of its classifications which more affect the performance of the recognizer. If the word spoken by the speaker is not present in the vocabulary list, then the system will not able to recognize the word and give OOV (Out of Vocabulary) rate error. According to the vocabulary size the system can be classified as small vocabulary size systems, medium vocabulary size systems and large vocabulary size systems.

1.4 Purpose –

The primary purpose of this research work is to understand about the development process of speech recognition system and existing tools for the development of an effective and high-performance open speech recognition API for minority languages.

1.5 Reason of Dying Minority Mother Tongues –

Due is English is an international language, there is the influence of English language everywhere. But the major reason in some countries or some areas specially in India, the education system focus on English language rather than to focus on the subject teaching.

1.6 Issues with existing Speech API's –

Dragon Naturally Speaking from Nuance is very efficient but not freely available, depends upon speaker to recognize and supports only seven languages. Windows Speech Recognition from Microsoft performs very effectively, can do almost all the things which can done by mouse and keyboard, supports 26 languages but needs a lot of training for effective performance. Google Docs Voice Typing from Google mainly used for typing Google Docs documents online but recognize very effectively without any training requirement before start using and supports 119 languages which are much more than Dragon & Windows Speech Recognition. But no Speech API support the Punjabi speech recognition. Only one Liv.ai API[3] support Punjabi voice input but its concept is based on transcription not on speech recognition. Due to that it has low accuracy, high response time and high not transcription rate.

2. LITERATURE REVIEW

Speech or voice input can be categorized as isolated words, connected words and continuous speech and automatic speech recognition system can be classified as: speaker mode, speech mode, speaking style and vocabulary size. Speech recognition techniques can be classified as: acoustic phonetic approach, pattern recognition approach and artificial intelligence approach. There are various factors like incomplete sentences, noise, body language, channel variability, speaker variability, speaking

style, pronunciation of same word more than once, age of speaker, speaking of mother language or any other language which becomes difficulties for an automatic speech recognition system during recognition.[2]

The concept of voice recognition in Google Search Engine (GSE) is based on Native method which are the Java method for combining the power of C or C++ with Java programming, Robot class in Java which takes the control over the mouse and keyboard commands by using voice commands which means it creates the Java methods for replacing the keyboard and mouse events with voice command events, grammar file which are able to recognize the voice input, Sphinx-4 a Java based framework in which Google Search Engine for voice recognition is developed and Wiener filter[4] for noise filtration which also explain about the commonly used commands and implementation.

Google's efforts for voice recognition being with GOOG-411 [1] a dial directory system for calling and finding business contacts. It's the initial effort and after that a new and improved version of GOOG-411 came in 2008. In this the output is only through speech. In March 2008 Google introduced a new service GMM (Google Maps for Mobile) by which the users can see the result on maps and in November 2008 Google introduce GMA (Google Mobile App) for iPhone which enable the users to use internet with voice searching. Bacchiaani give detailed explanation of GOOG-411 in [5].

Google performs experiments on its voice search by using distributed discriminative language model[6] which is ubiquitous in nature and implemented efficiently using MapReduce framework. The training of these languages model is performed by using a distributed perceptron algorithm (N-gram). These types of language models require large amount of data for efficiency. The experiment performed consists of 27,273 utterances or 87,360 words & manually transcribed.

The aim of WSR is to design a simple and intelligent framework where user remain in the control of system at all the time. The main elements of WSR are: the speech user experience, the speech application programming interface (SAPI), the speech recognition engine (SRE), the speech synthesis engine and the audio subsystem.[7] After producing of recognition engines & speech API's, Microsoft team releases SAPI 5 SDK which includes both text to speech and speech to text controls.

Variable sliding window dynamic time wrapping (DTW) algorithm which is used for identification of speech of variable length of speech by using template matching approach and based on dynamic programming.[8] Its widely used than HMM (Hidden Markov Model) because of same conditions the results of both algorithm are same but HMM is complex than variable sliding window DTW algorithm due to number of iterations needed.

3. EVALUATING DEVELOPMENT FACTORS OF EXISTING SPEECH API'S IN CONTEXT OF ARCHITECTURAL COMPARISON

3.1 Google Voice Typing –

- The Google voice typing is implemented using Java programming language which is more flexible, portable, simple and secure by using Sphinx-4 as development tool which is also written in Java.[4]
- Because Sphinx-4 does not include any noise reduction module so Google voice typing use Wiener filter to filter the noise. [4]
- The Google voice typing concept stand on native method, robot class, grammar file, sphinx-4 and Wiener filter which are very well defined in [4].

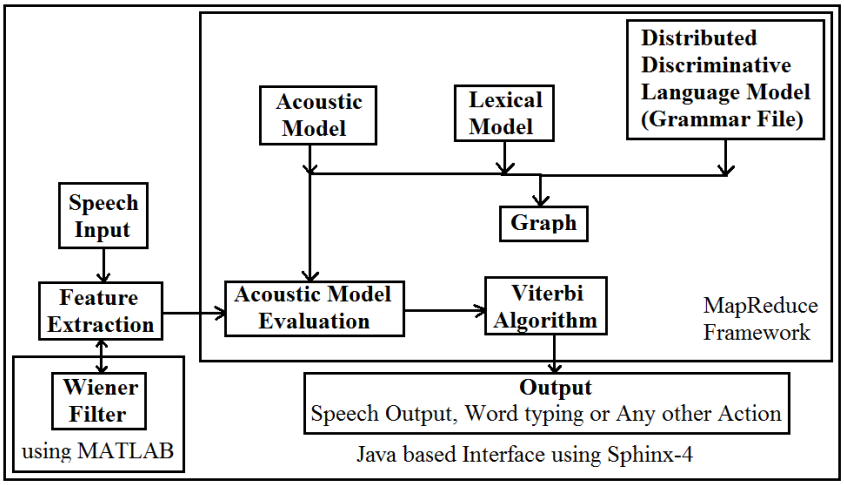

Figure 3: Basic Architecture of Google Voice Typing [1]

- MATLAB is used to test the noise reduction methods using Wiener filter.[4]
- Native Methods are Java methods (class or instance) which combine the power of C or C++ with Java programming and enhance the interaction between program and operating system. [4]
- Robot class takes the control over the mouse and keyboard command by using voice commands.
- GOOG-411 telephony service was used for acoustic modeling which is based Hidden Markov and Gaussian Mixture Models[1], [5].

- Viterbi algorithm is used for quantization and pruning.[9]
- MapReduce programming model is used for optimizing the results using Discriminative n-gram language model.[6]
- Google Cloud SQL is used as speech database to deal with big data analytics.[10], [11]
- Metrics of Google voice search includes Word Error Rate (WER), Semantic Quality (WebScore), Perplexity (PPL), Out-Of-Vocabulary Rate (OOV) and Latency which are very well defined in [1].
- It is available only for Google Chrome web browser which is available for Windows, Mac OS, Linux, Android and iOS operating systems.[19] It's a build in feature of Chrome browser. Hence Google Docs Voice Typing is platform independent system.
- It's basically used to write Google Docs documents by voice on desktops. Its Voice Search app onandroid can be used to do most of the things which you want to do.

3.2 Windows Speech Recognition –

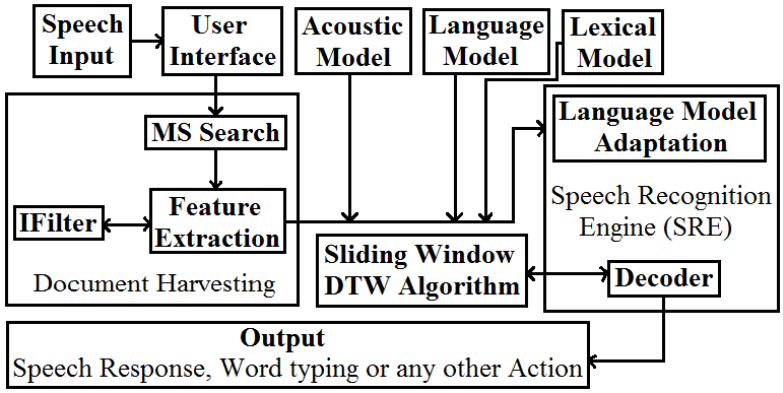

Figure 4:Basic Architecture of Windows Speech Recognition[7]

- The key elements of Windows speech recognition architecture include the speech UX (the speech user experience), the speech API, the speech recognition engine (SRE), the speech synthesis engine, the audio subsystem. [7]
- Because the Windows Speech Recognition is implemented using .NetFramework 3.0 so its development language must be visual basic or C#. For development its speech recognition engine (SRE) includes the elements from both Sphinx and Entropic recognizers.[7]
- To enable the documents searchable and readable from various file formats, text conversion and for indexing them IFilter plug in is used which is the combination of macro and micro level.[7]
- Hidden Markov Model and Gaussian Mixture Model is used to match the input speech data and sample speech data i.e. for acoustic modeling.[7]
- A variable sliding window DTW (Dynamic Time Warping) algorithm which is based on dynamic programming is used to extract parameters and to make language model more robust.[7], [8]
- The SAPI (Speech Application Programming Interface) of windows speech recognition use VB Active X programming model.[13]
- Unigrams, bigrams and trigrams are used for language modeling.[7]
- MSSearch service of SQL server is used for querying and retrieving information from speech database interface.[7]
- Matching mode is used to accept the subset of spoken word when user not use the full name of application and spelling model is used to remove disambiguation.[7]
- As its name suggests it only supports Windows. Its available for Windows Vista, 7, 8, 8.1& 10. Hence its platform dependent system.
- It's basically used to control application software's on your desktop and to navigating desktop by giving voice commands. It can do all most everything which we do with keyboard and mouse.

3.3 Dragon Naturally Speaking –

- Because the Dragon is a paid software so not much information is available for research. According to the best knowledge the Dragon is able to dictate 160 words per minute[21] and three times faster than typing.
- It's comes under speaker-dependent speech recognition system.[18]
- The probabilistic function of Markov process(Hidden Markov Model and algorithm) is used to find the optimal path to recognize the input utterance and for pattern matching.[12]
- The features of Windows Speech Recognition and DNS are very similar. Dragon Naturally Speaking basically used for browse the web, controlling applications and desktop, for recording and writing documents which described at [20].

– DNS require special system requirements. Without these requirements Dragon system will not work properly.[22]

Table 1: ASR Tools Comparison

Evaluation Parameters	Google voice typing	Windows Speech Recognition	Dragon Naturally Speaking
Development Tool	Sphinx-4[4]	Sphinx and Entropic[7]	———
Programming Language	Java Programming Language[4]	VB	———
Simulator	MATLAB[4]	_____	———
Acoustic Model	Hidden Markov Model and Gaussian Mixture Model[1], [5]	Hidden Markov Model and Gaussian Mixture Model[7]	Hidden Markov Model[12]
Filter	Wiener Filter[4]	IFilter[7]	_____
Language Model	n-grams DML[6]	n-grams[7]	———
Programming Model	MapReduce[6]	VB Active X[13]	———
Speech Database	Google Cloud SQL[10], [11]	MSSearch(SQL)[7]	———
Algorithm	Viterbi[14]	Sliding Window DTW[7], [8]	Hidden Markov Algorithm[12]
Languages Available	119 languages[15]	26 languages[16]	7 languages[17]
Classification	Speaker-independent	Speaker-independent	Speaker-dependent[18]
Speech API	Google Speech API	Microsoft Speech API	Dragon Speech API
Platform	Windows, Mac, Linux, Android & iOS.[19]	Windows Vista, 7, 8, 8.1, 10.	Windows 7, 8, 8.1, Windows Server 2008 R2, Windows Server 2012
Usability	Writing Google Docs Documents	Typing, web browsing, controlling & navigating desktop.	Typing, web browsing, controlling & navigating desktop, recording.[20]

4. PROPOSED MODEL

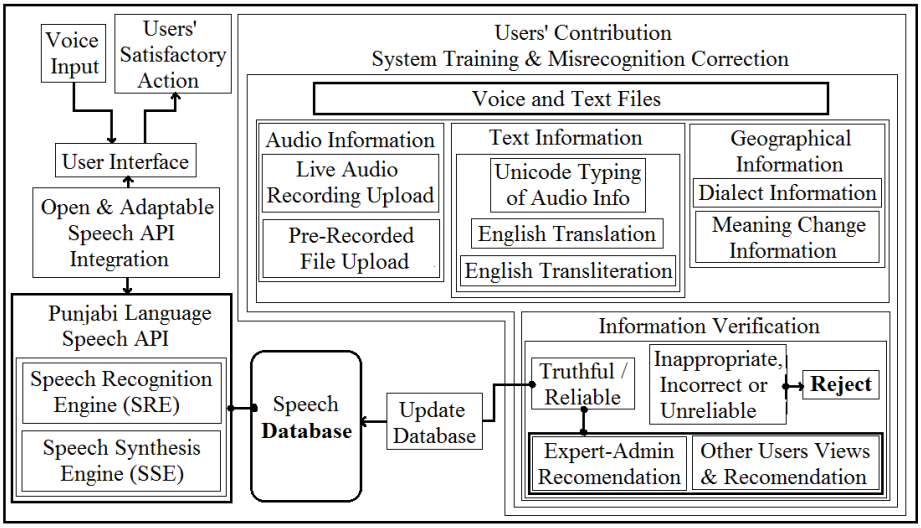

Figure 5: MMT Speech API

4.1 Proposed Likely Results &Benefits over Current –

By the users' contribution and information addition to database only with expert-admin recommendation the system will self-enriching with truthful information. Due to its open system, any developer can easily integrate it with own software, application or website and can also able to train it into own mother tongue by adding new mother tongue or give more

training to improve accuracy if system already has that mother tongue. The information provided by one user is also visible to other users by which any user can comment on that and offer improvements. By the geographical information the system will capable to accurately recognize the dialect changing voice input. It will perform a glowing role to publicize, preservation, regeneration and promotion of dying minority mother tongues.

5. EXPERIMENTS

Experiment 1– Plotting a live audio signal using MATLAB 7.1 [23]

This code records the live audio up to 60 seconds.

```
>> R = audiorecoder;
>> recordblocking(R,60);
>> RecordedData = getaudiodata(R);
>> plot(RecordedData);
```

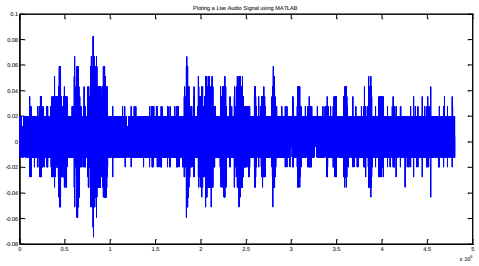

Figure 6: Two-dimensional plotting live audio signal.

Experiment 2– Plotting a previously recorded audio signal (.wav file) using MATLAB 7.1[24]

```
>> [x,fs] = wavread('D:\audio.wav');
>> t = [1/fs:1/fs:length(x)/fs];
>> plot(t,x);
```

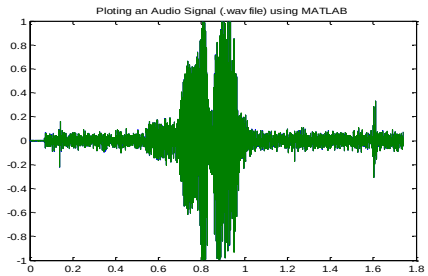

Figure 7: Two-dimensional plotting of .wav file.

Figure 7: Two-dimensional plotting of .wav file.

6. CONCLUSION

A speech API system is proposed for the preservation, advertisement, publication & regeneration and rise the use of basic words of Punjabi language. It is an open system in which user can participate to add words and can train the system to mold into own necessity. Users' contribution is the important module of proposed API those purpose is to become system self-enriching. The foremost features are better accuracy, user accordingly action, desirable response time, self-enrichment, portable and open system. The desirable response time is satisfactory for self-enriching database. After the basic structure it's a continuous process of improvement for accuracy and response time. Due to speaker-independence, no training requirement, more languages and totally Java based environment which is more flexible, portable, open and robust, Google Speech API is the best option for exploration in further research.

7. FUTURE SCOPE

It's a continuous process to large and enrich speech database with truthful information. There are various versions of Punjabi language not only according to different countries and states but also in the smaller areas of same state. Due to this lot of dialect variation are there. The system needs to focus on pronunciation variations. The accuracy will improve if user select the voice input according to a specific area and system is capable to show various recognitions according to similar dialect in the case of misrecognition. Its better a continuous commenting section in which user give comments with audio and text file like "I am speaking this… but system is recognizing this…".

8. REFERENCES

[1] J. Schalkwyk et al., "Google Search by Voice : A case study," pp. 1–35.

[2] S. J. Arora and R. P. Singh, "Automatic Speech Recognition : A Review," Int. J. Comput. Appl., vol. 60, no. 9, pp. 34–44, 2012.

[3] https://liv.ai/.

[4] H. Al_bool, M. Ali, H. Ibrahim, and A. M. M. Habbal, "Hands-Free Searching Using Google Voice," Int. J. Comput. Sci. Netw. Secur., vol. 10, no. 8, pp. 47–53, 2010.

[5] M. Bacchiani, F. Beaufays, J. Schalkwyk, M. Schuster, and B. Strope, "Deploying GOOG-411: Early lessons in data, measurement, and testing," ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc., pp. 5260–5263, 2008.

[6] P. Jyothi, L. Johnson, C. Chelba, and B. Strope, "Distributed Discriminative Language Models For Google Voice-Search," Word J. Int. Linguist. Assoc., pp. 5017–5020, 2012.
[7] J. Odell and K. Mukerjee, "Architecture, user interface, and enabling technology in Windows Vista's speech systems," IEEE Trans. Comput., vol. 56, no. 9, pp. 1156–1168, 2007.
[8] G. Kang and S. Guo, "Variable sliding window DTW speech identification algorithm," 2009 9th Int. Conf. Hybrid Intell. Syst. HIS 2009, vol. 1, pp. 304–307, 2009.
[9] X. Gonzalvo, S. Tazari, C. Chan, M. Becker, A. Gutkin, and H. Silen, "Recent Advances in Google Real-time {HMM}-driven Unit Selection Synthesizer," pp. 2238–2242, 2016.
[10] https://cloud.google.com/speech/.
[11] https://cloud.google.com/sql/faq#whatissql.
[12] J. K. Baker, "The Dragon System-An Overview," IEEE Trans. Acoust. Speech Signal Process., vol. 23, no. 1, pp. 24–29, 1975.
[13] K. A. Jones, Windows speech recognition programming : with Visual Basic and ActiveX voice controls. iUniverse, Inc, 2004.
[14] X. Gonzalvo, S. Tazari, C. Chan, M. Becker, A. Gutkin, and H. Silen, "Recent Advances in Google Real-time HMM-driven Unit Selection Synthesizer," Interspeech, pp. 2238–2242, 2016.
[15] https://support.google.com/docs/answer/4492226?hl=en.
[16] https://msdn.microsoft.com/en-us/library/hh378476(v=office.14).aspx.
[17] http://nuance.custhelp.com/app/answers/detail/a_id/6280/~/what-languages-are-available-for-dragon-naturallyspeaking-11%3F.
[18] "Dragon Naturally Speaking White Paper," no. March, pp. 1–10, 2009.
[19] http://searchmobilecomputing.techtarget.com/definition/Google-Chrome-browser.
[20] http://www.dummies.com/software/dragon-naturallyspeaking/what-dragon-naturallyspeaking-can-do-for-you/.
[21] "Dragon NaturallySpeaking 10 SDK Client Edition."