ISSN: 2348 - 2117

International Journal of Engineering Technology and Computer Research (IJETCR)
Available Online at www.ijetcr.org
Volume 2; Issue 7; Page No. 01-06

Optimized Move Toward to Voice Translation

Nitesh Patel[1], Prof.V.N.Patil[2]

[1]PE'S Modern College of Engineering, Shivajinagar, Pune, Maharashtra, India-410011

Niteshpatel.dyp@gmail.com

[2] PE'S Modern college of Engineering, Shivajinagar, Pune, Maharashtra, India-410011

Vvnnpp2002@gmail.com

ABSTRACT

Current voice conversion tools and services use natural language understanding and natural language processing to change words. However, these parsing methods focus more on capturing keywords and translating them, completely neglecting the significant quantity of processing time concerned. In this paper, we are suggesting technique that can optimize the processing time thereby increasing the throughput of voice conversion services. Techniques like template matching, indexing frequently used words using likelihood search and session-based cache can considerably enhance handing out times. More so, these factors become all the more important when we need to achieve actual-time conversion on computers, movable handsets or laptops

Key Words: Conversion, language, Template matching, Natural Processing Language,Spinx

INTRODUCTION:

We present the design of movable handset users who can converse with other users using software model that ensures effective actual time communication between two users who do not speak a common language.[1] The voice is establishedd at first and then translated to the target language which is then translated to speech. The motivation for working and understanding the project requirements and the usefulness had been thoroughly done. Easy for other in understanding many parts in the project had been present. The main role in successful completion and running of the project had been due to my guide Prof. V.N.Patil. With his dedication and always available for guidance have boosted mind to work for better of the project.

SCENARIOS

Our aim is at movable handset users who can converse with other users using software model that ensures effective actual time converse between two users who do not speak a common language.[1] The voice is established at first and then translated to the target language which is then translated to vocalizations.

MOTIVATION

The motivation for working and sympathetic the project requirements and the usefulness had been thoroughly done. Easy for other in understanding many parts in the project had been present. The main role in successful completion and running of the project had been due to my guide Prof. V. N. Patil. With his dedication and always available for guidance have boosted mind to work for better of the project.

PROPOSED IMPLEMENTATION

During communication it is essential to established a language correctly. Since many times acknowledgment is not just the goal, the result of acknowledgment is an intermediate of the system. The result is further used as input for another module. Therefore if the task of established is not correct then modules which rely on the result of established may not perform the further operation correctly or may produce partially correct result.

VOICE TO TEXT CONVERSION SOFTWARE

Researchers at Microsoft have made software that can learn the sound of our voice, and then use it to speak a language that you don't. The system could be used to make language tutoring software more personal, or to make tools for travelers. The new technique could also be used to help students learn a language, said Soong. Providing sample foreign phrases in a person's own voice could be encouraging, or easier to imitate. Soong also showed how his new system could improve a navigational directions phone app, allowing a stock synthetic English voice to seamlessly read out text written on Chinese road signs as it relayed instructions for a route in Beijing.

SYSTEM ANALYSIS

A system is characterized by how it responds to input signals. In general, a system has one or more input signals and one or more output signals. We use MISO (Multiple Inputs, Single Output). Our software will provide easy to use graphical user interface (GUI). It will provide option to select source language. It will established the language and display the spoken words and then go for further process.

TECHNOLOGIES USED

Eclipse, a java based platform. Sphinx, an application programming interface for conversion voice to text. TTS, an application programming interface used for conversion of text to voice. Dhwani, a software used to speak the output in text in language selected. Ubuntu 12.10, an OS installed at server side used to support and run Dhwani software. Microsoft word, used to store the pronounciations of the different words used and a dictionary file. Windows 7, an OS installed at client side.

SYSTEM ARCHITECTURE

The voice conversion model consists of four main components namely: speech established, natural language interpretation and analysis, sentence generation and text-to-speech synthesis. The optimization is provided by the natural language interpretation and analysis module is which further divided into four parts namely: template matching, indexing frequently used words, session-based cache and conversion to target language. Figure 2 depicts the overall system model for optimization of voice conversion on movable handsets [1].

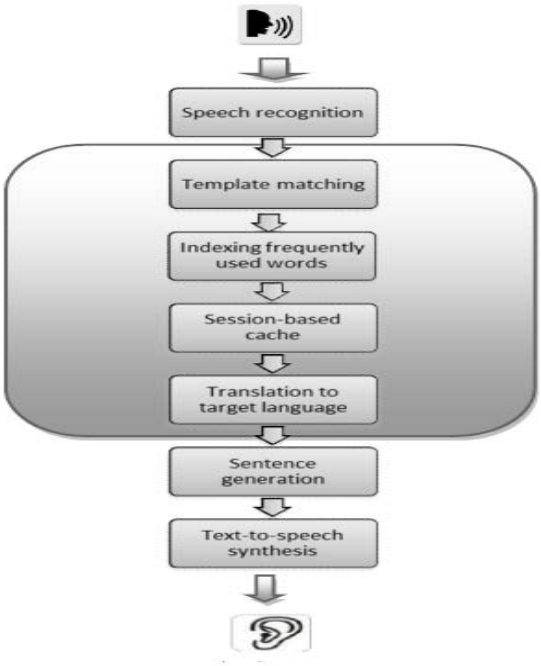

Figure 1: System Model [1]

SPEECH ESTABLISHED COMPONENT

Speech established is the process of converting an acoustic signal captured by the mobile's microphone to a set of meaningful words. Speech established systems can be characterized by many parameters like speaking mode, speaking style, speaking accents and signal-to-noise ratio. It takes into consideration the speaking accents of the caller. Established is more difficult when vocabularies are large or have many similar-sounding words. To implement this speech established module, Sphinx, a speech established system is used. It is written entirely in the JavaTM programming language.[1]

TEMPLATE MATCHING

Template matching checks the source input for commonly used phrases or sentences. Every language consists of a set of commonly spoken words or sentences. Converting such sentences to the target language and replacing when required drastically reduces the processing time needed for translating the sentences. Consider the statement "How are you?" as a commonly used sentence. The translated text output of this sentence is stored in a relational table. If a person speaks this sentence, then it will be directly translated instead of disassembling the words and analyzing them. [1]

INDEXING FREQUENTLY USED WORDS

The large lexical database of words will add to the time complexity of the process. The words are indexed based on the number of times a particular word has been used. The probability search algorithm is used to index the words in the database. In probability search the most probable element is brought at the beginning. When a key is found, it is swapped with the previous key. Thus if the key is accessed very often, it is brought at the beginning. Thus the most probable key is brought at the beginning. The efficiency of probability search increases as more and more words are being translated and indexed. The presence of a single while loop, makes the time complexity of this algorithm as O(n). The best case will be when the word to be searched is at the beginning while worst case will be when the word is at the end [1].

SESSION-BASED CACHE

The system maintains a session-based cache for each user requesting for the service. This works on the lines of a web cache which caches web pages. This is done to reduce to reduce bandwidth usage, server load, and perceived lag. It is assumed that when a user engages in a conversation, there are bound to be multiple repetitions of certain words. Based on this assumption, we cache such words along with their translated text so that server processing time is saved. [1]

CONVERSION TO TARGET LANGUAGE

After the sentence passes through the first three phases of language interpretation and analysis, the final phase is conversion to target language. [1]

SENTENCE GENERATION

The collective set of translated word is converted to a meaningful sentence generation for the target language. Sentence generation is a natural language processing task of generating natural language from a logical form (set of translated words). [1]

TEXT-TO-SPEECH SYNTHESIS

Text-to-speech synthesizer converts the sentence obtained from the sentence generation module into human speech form in the trget language. This is done by using freeTTS software .[1] [3]

DATA FLOW DIAGRAMS

This diagram focuses explicitly on the data exchanges within the system, with no notion of control.

1. LEVEL 0

In this, the user one will call user two using voice conversion program using the system. The user one will speak in his language and press the button to which he wants to translate his message to user two. After doing this the user two will receive the message understood by user two in his language. This is how the communication is done.

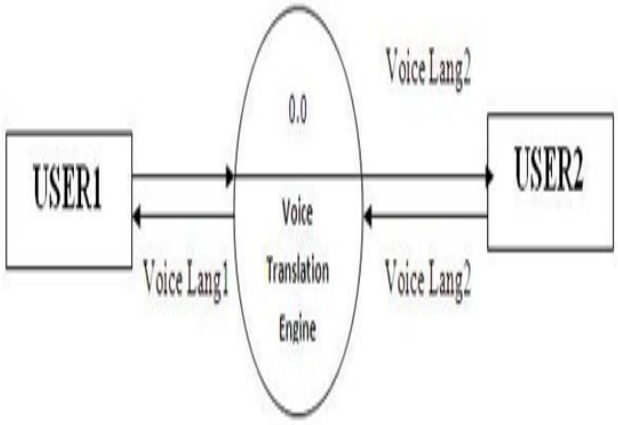

Figure 2: Level 0

2. LEVEL 1

In this, user1 will speak in his language and on the voice conversion system. The voice conversion system has components like voice established, voice conversion engine and text to speech. The voice established machine will established the voice and search for the related words in its database. These translated words are then converted into a text file and given o voice conversion machine. It will translate the text into another translated text that is the output of user1 conversion. This text file is given to text to speech component and the text file is converted into speech.

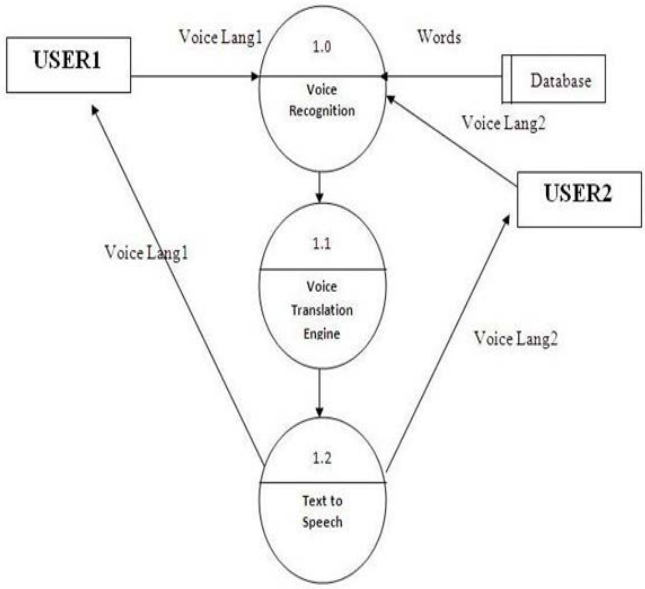

Figure 3: Level 1

3. LEVEL 2

In this we on the voice conversion program which in turn will call the Google API. This Google API will take the words and convert it into text file. This text file is then given to the text to speech convertor. This speech is given to voice synthesizer module which speaks in the language understood by the receiver.

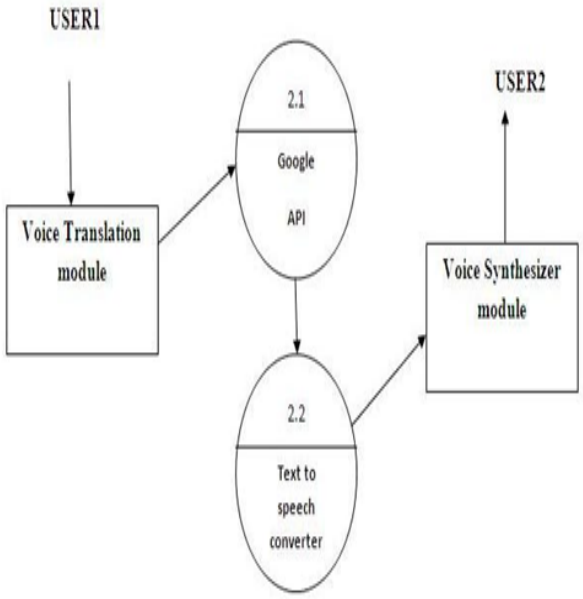

Figure 4: Level 2

BASIC ACTIVITY DIAGRAM

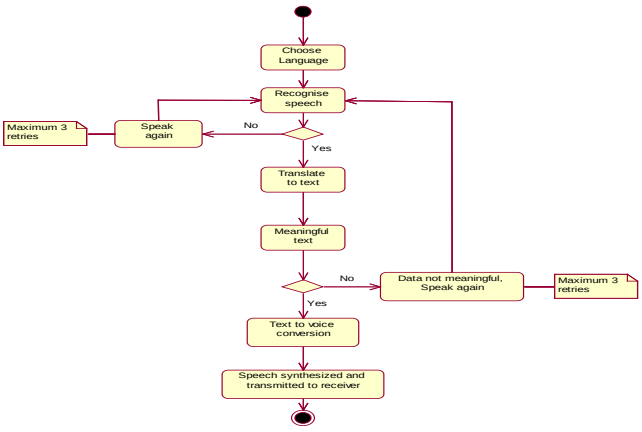

Figure 5: Activity Diagram

Choose language to be translated. This translated language will established the speech if it is not established then speak again and maximum tries given for established are three. If it establisheds the conversion then convert speech to meaningful text if the text is meaningful then convert it again back to voice and speech synthesizer will transmit back to receiver. If it not meaningful then speak again and the maximum tries are three.

B. Sequence Diagram

User will select the language and make a call to the speech establishedr system. The speech establishedr system will established the call and translate the speech to text. Establishedr will extract the keywords. The language translator system will convert it to appropriate language and make a sentence and convert it to voice string. Speech synthesizer will give acknowledgement to the receiver.

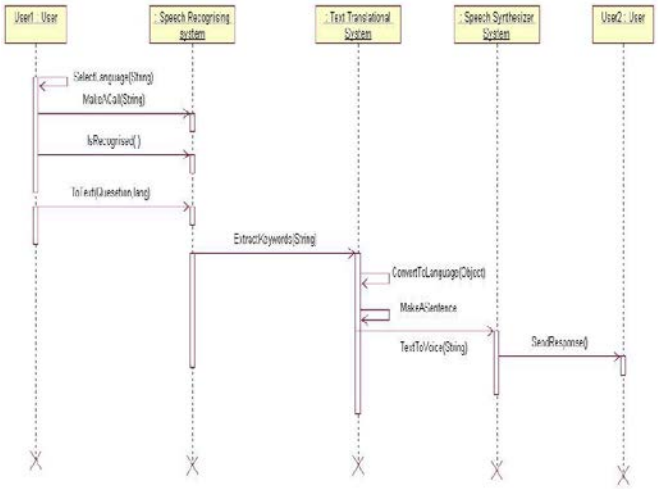

Figure 6: Sequence Diagram

C. Component Diagram

In this we have three components a mobile client, voice conversion system and mobile client receiver. Voice translator will perform three action it will convert speech to translated text, text to the text that translated and then translated text to the speech which the receiver client wants to hear.

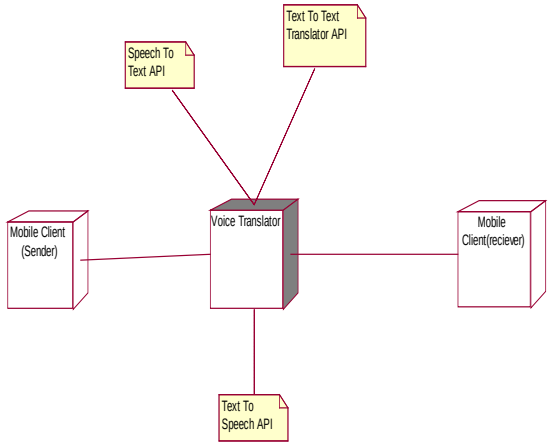

Figure 7: Component Diagram

IMPLEMENTATION DETAILS

A. Related Algorithms For Voice Established & Conversion - Hidden Markov Models

Modern general-purpose voice conversion systems are based on Hidden Markov Models. These are statistical models that output a sequence of symbols or quantities. HMMs are used in speech established because a speech signal can be viewed as a piecewise stationary signal or a short-time stationary signal.. Speech can be thought of as a Markov model for many stochastic purposes. Another reason why HMMs are popular is because they can be trained automatically and are simple and computationally feasible to use. In speech established, the hidden Markov model would output a sequence of *n*-dimensional actual-valued vectors (with *n* being a small integer, such as 10), outputting one of these every 10 milliseconds. The vectors would consist of cepstral coefficients, which are obtained by taking a Fourier transform of a short time window of speech and decorrelating the spectrum using a cosine transform, then taking the first (most significant) coefficients. The hidden Markov model will tend to have in each state a statistical distribution that is a mixture of diagonal covariance Gaussians, which will give a likelihood for each observed vector. Each word, or (for more general speech established systems), each phoneme, will have a different output distribution; a hidden Markov model for a sequence of words or

phonemes is made by concatenating the individual trained hidden Markov models for the separate words and phonemes. Decoding of the speech (the term for what happens when the system is presented with a new utterance and must compute the most likely source sentence) would probably use the Viterbi algorithm to find the best path, and here there is a choice between dynamically creating a combination hidden Markov model, which includes both the acoustic and language model information, and combining it statically beforehand (the finite state transducer, or FST, approach).

The Hidden Markov Model (HMM) is a variant of a *finite state machine* having a set of hidden *states*, *Q*, an output *alphabet* (observations), *O*, transition probabilities, *A*, output (emission) probabilities, *B*, and initial state probabilities, Π. The current state is not observable. Instead, each state produces an output with a certain probability (*B*). Usually the states, *Q*, and outputs, *O*, are understood, so an HMM is said to be a triple, (*A*, *B*, *Π*).

Hidden states $Q = \{ q_i \}$, $i = 1, \ldots, N$. Transition probabilities $A = \{a_{ij} = P(q_j \text{ at } t+1 \mid q_i \text{ at } t)\}$, where $P(a \mid b)$ is the conditional probability of *a* given *b*, $t = 1, \ldots, T$ is time, and q_i in *Q*. Informally, *A* is the probability that the next state is q_j given that the current state is q_i.

Observations (symbols) $O = \{ o_k \}$, $k = 1, \ldots, M$.

Emission probabilities $B = \{ b_{ik} = b_i(o_k) = P(o_k \mid q_i) \}$, where o_k in *O*. Informally, *B* is the probability that the output is o_k given that the current state is q_i.

Initial state probabilities $\Pi = \{p_i = P(q_i \text{ at } t = 1)\}$.

The model is characterized by the complete set of parameters: $\Lambda = \{A, B, \Pi\}$.

B. Flow Chart For Voice Conversion Menu

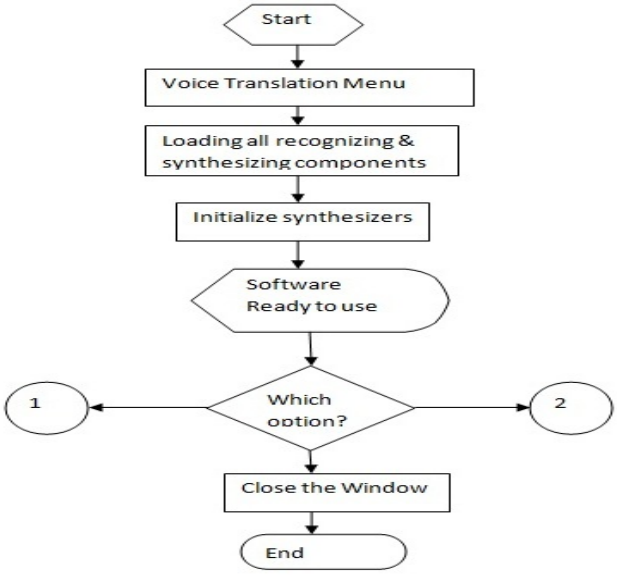

Figure 9: Flow chart for Voice Conversion menu

In this diagram, the process starts with process voice conversion menu. Then loading all recognizing and synthesizing component process begins. After this process, synthesizers are initialized and a message is displayed that the software is ready to use. Then user has to select the option from the options given on the menu i.e. English to Hindi conversion or Hindi to English conversion of language or close the window. If user selects English to Hindi conversion the control goes to connector named as 1.If user selects Hindi to English

RESULTS AND DISCUSSION

1. Voice (In English) Established At Client Side

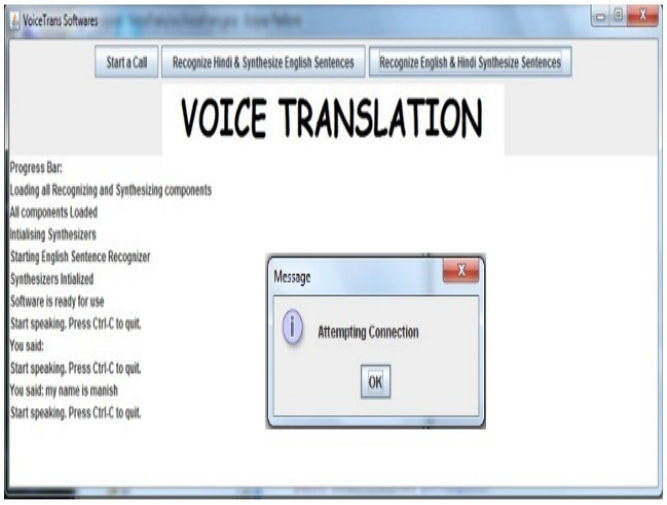

Figure 10: Voice (in English) established at client side

2. Output Voice (In Hindi) At Server Side

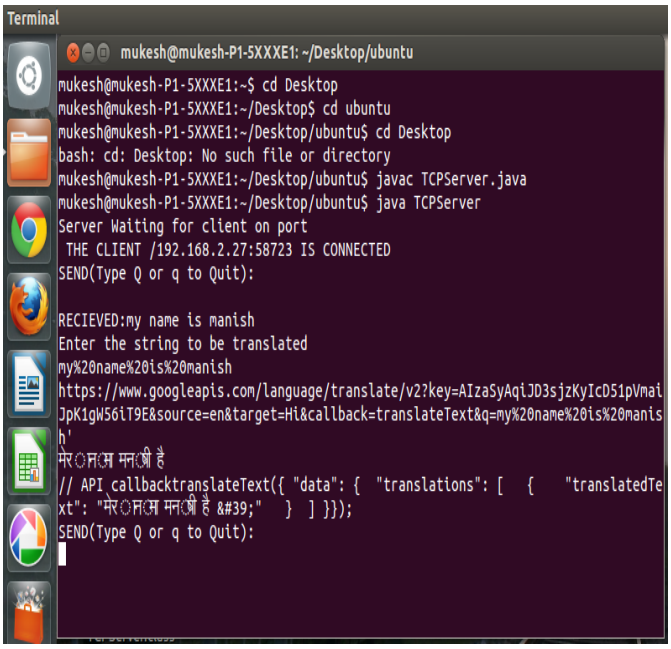

Figure 11: Output voice (in Hindi) at server side

3. VOICE (IN HINDI) ESTABLISHED

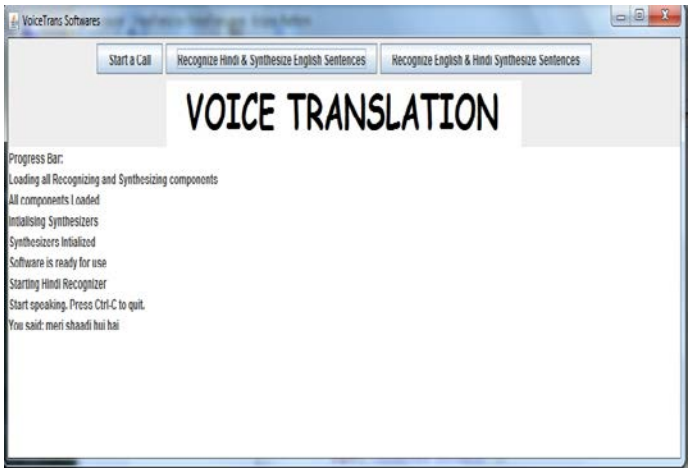

Figure 12: Voice (in Hindi) established

5.3.4 VOICE (IN ENGLISH) OUTPUT

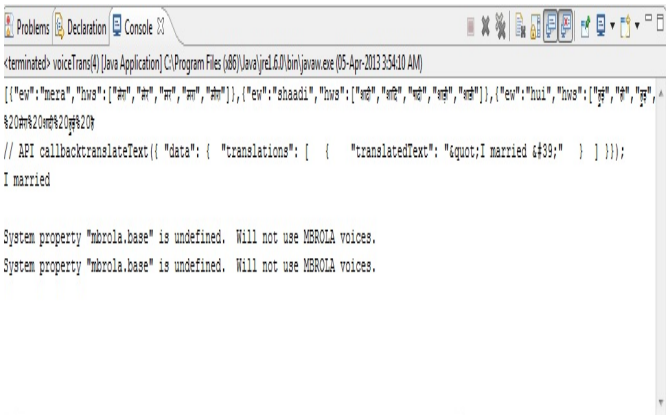

Figure 13: Voice (in English) output

CONCLUSION

Language has always been a barrier to effective communication. As businesses expand and technology engulfs the entire globe, reliable and actual-time conversion becomes imperative. While considerable progress has been done in this direction, more efforts need to be taken in order to reduce the enormous processing time involved with it. With this project, we have developed a new system model to provide actual-time communication between two users who do not speak a common language.

REFERENCES:

1. Yen Chun Lin “An optimized approach to voice conversion on movable handsets”, 2010.
2. Titus Flex FORTUNA ” Dynamix Programming Algorithms in Speech established”, 2008,pp 94-99.
3. Srinivas Banglore, Vivek Kumar Rangarajan Sridhar, Prakash KolanLadan Golipur, Aura Jimenez ”Actual-time Incremental Speech-to-Speech Conversion of dialogs”, 2012,Conference of North American Chapter of the association for computational linguistic ; Human Language Technologies, pp 437-445.
4. Willie Walker, Paul Lamere, Philip Kwok “Free TTS- A performance case