International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Optimized Approach to Voice Translation

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

[1]203, Rosemary Appt, Umang Society, Behind BSNL office, Talegaon Dabhade Taluka Maval Dist: Pune Maharashtra

Abstract: ***Current voice translation tools and services use natural language understanding and natural language processing to convert words. However, these parsing methods concentrate more on capturing keywords and translating them, completely neglecting the considerable amount of processing time involved. In this paper, we are suggesting techniques that can optimize the processing time thereby increasing the throughput of voice translation services. Techniques like template matching, indexing frequently used words using probability search and session-based cache can considerably enhance processing times. More so, these factors become all the more important when we need to achieve real-time translation on computers, mobile phones or laptops.***

Keywords: Real time translation on mobile phone, voice to text, text of one language to text of other, Translate

1. Introduction

We present the design of mobile phone users who can communicate with other users using software model that ensures effective real time communication between two users who do not speak a common language.[1] The voice is recognized 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.

2. System Design

A. Scenarios

Our aim is at mobile phone users who can communicate with other users using software model that ensures effective real time communication between two users who do not speak a common language.[1] The voice is recognized at first and then translated to the target language which is then translated to speech.

B. Motivation

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.

C. Proposed Implementation

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

D. Voice to Text Translation 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.

E. 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 recognize the language and display the spoken words and then go for further process.

F. Technologies Used

Eclipse, a java based platform. Sphinx, an application programming interface for translation voice to text. TTS, an application programming interface used for translation 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 pronunciations of the different words used and a dictionary file. Windows 7, an OS installed at client side.

3. System Architecture

The voice translation model consists of four main components namely: speech recognition, 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 translation to target language. Figure 2 depicts the overall system model for optimization of voice translation on mobile phones [1].

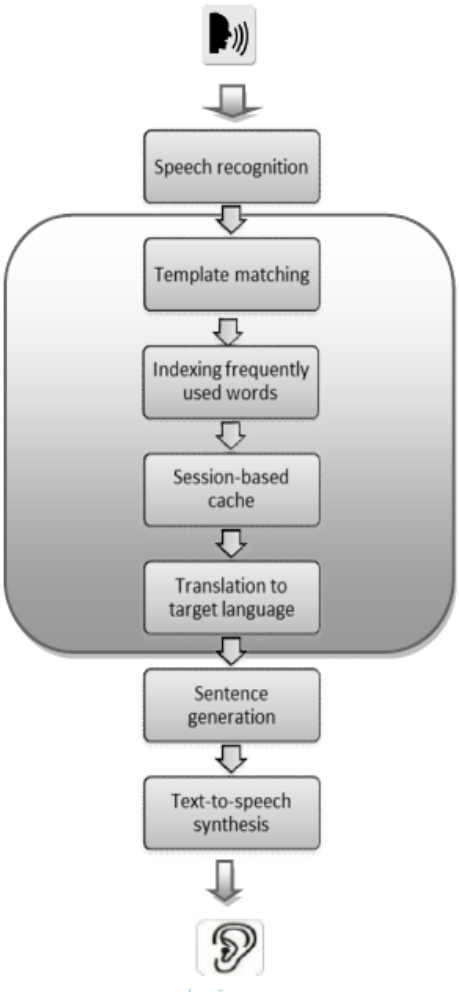

Figure 1: System Model [1]

a) Speech Recognition Component
Speech recognition is the process of converting an acoustic signal captured by the mobile's microphone to a set of meaningful words. Speech recognition 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. Recognition is more difficult when vocabularies are large or have many similar-sounding words. To implement this speech recognition module, Sphinx, a speech recognition system is used. It is written entirely in the Java™ programming language.[1]

b) 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]

c) 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].

d) 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]

e) Translation To Target Language
After the sentence passes through the first three phases of language interpretation and analysis, the final phase is translation to target language. [1]

f) 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]

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

4. 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 translation 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.

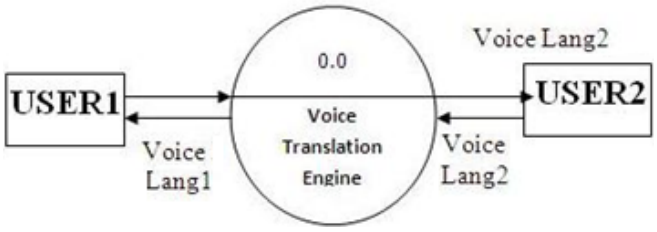

Figure 2: Level 0

2. LEVEL 1

In this, user1 will speak in his language and on the voice translation system. The voice translation system has components like voice recognition, voice translation engine and text to speech. The voice recognition machine will recognize 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 translation machine. It will translate the text into another translated text that is the output of user1 translation. This text file is given to text to speech component and the text file is converted into speech.

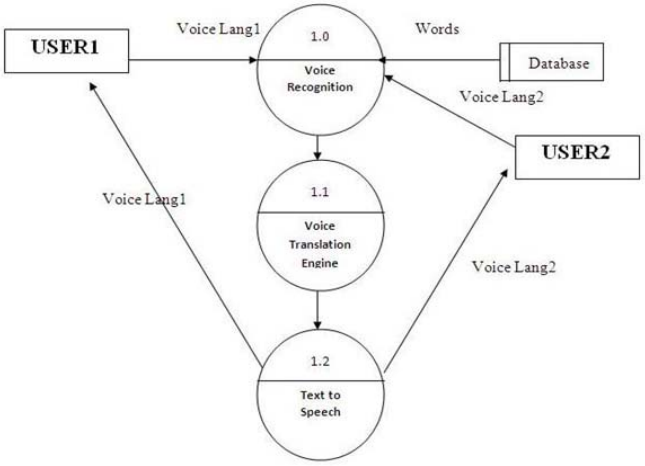

Figure 3: Level 1

3. LEVEL 2

In this we on the voice translation 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.

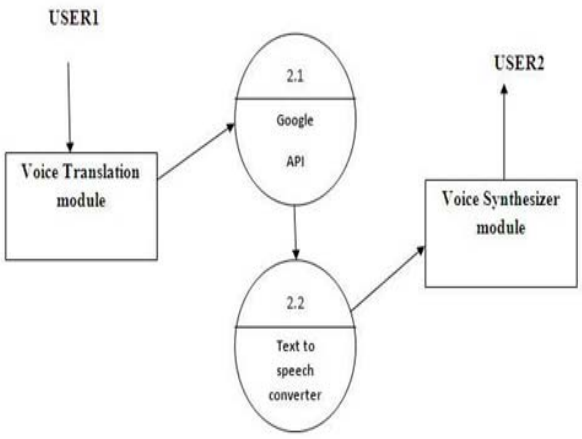

Figure 4: Level 2

5. Activity Diagram

5.1 Basic Activity Diagram

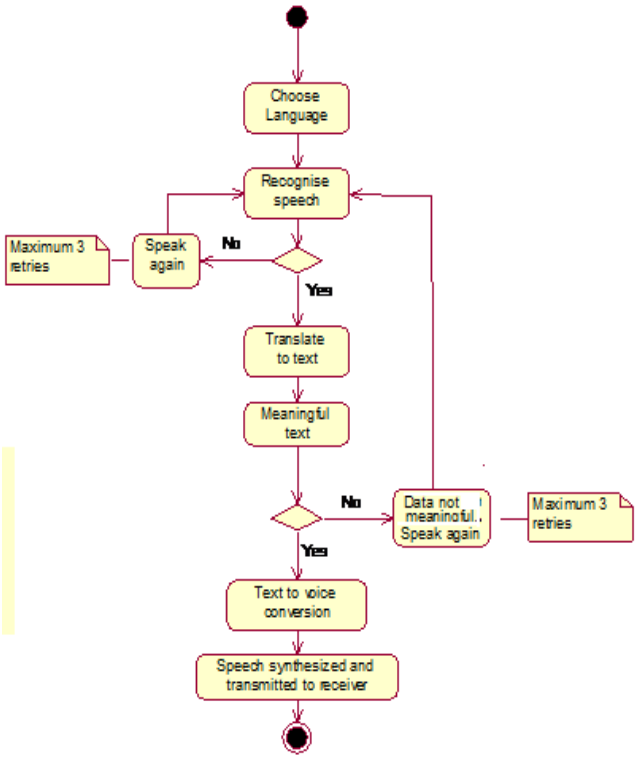

Figure 5: Activity Diagram

Choose language to be translated. This translated language will recognize the speech if it is not recognized then speaks again and maximum tries given for recognition are three. If it recognizes the translation 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 speaks again and the maximum tries are three.

5.2 Sequence Diagram

User will select the language and make a call to the speech recognizer system. The speech recognizer system will recognize the call and translate the speech to text. Recognizer 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.

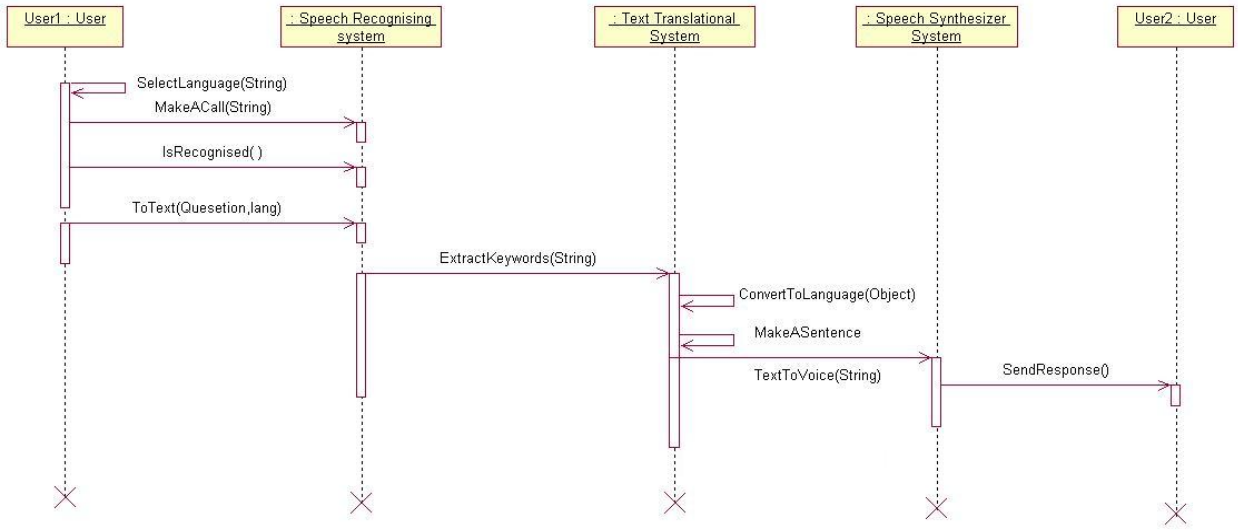

Figure 6: Sequence Diagram

5.3 Component Diagram

In this we have three components a mobile client, voice translation system and mobile client receiver. Voice translator will perform three actions 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.

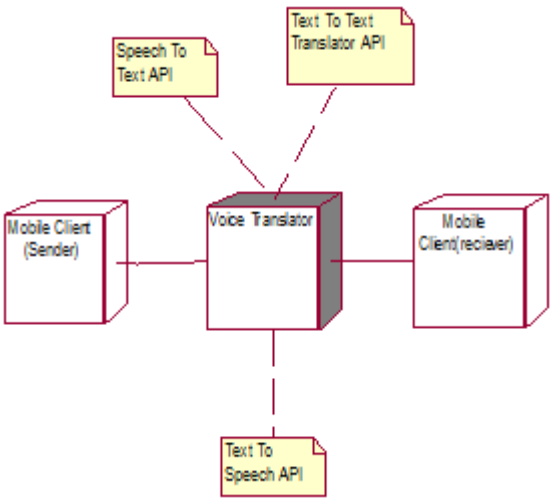

Figure 7: Component Diagram

6. Implementation Details

6.1 Related Algorithms for Voice Recognition & Translation - Hidden Markov Models

Modern general-purpose voice translation systems are based on Hidden Markov Models. These are statistical models that output a sequence of symbols or quantities. HMMs are used in speech recognition 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 recognition, the hidden Markov model would output a sequence of n-dimensional real-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 likelihood for each observed vector. Each word, or (for more general speech recognition 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 $\Pi = \{p_i = P(q_i \text{ at } t = 1)\}$.

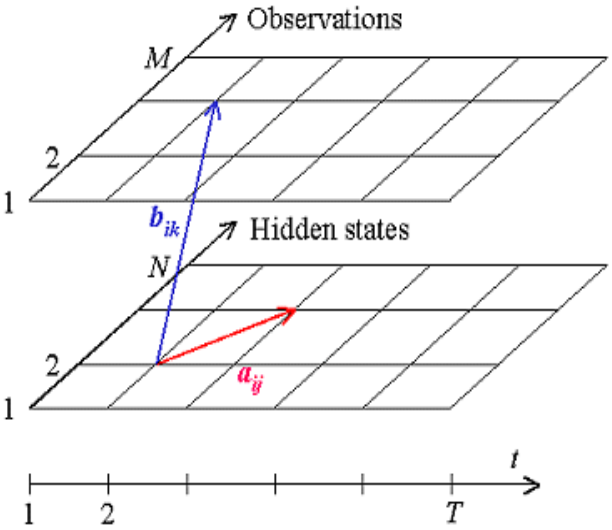

Figure 8: Hidden Markov model

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

6.2 Flow Chart for Voice Translation Menu

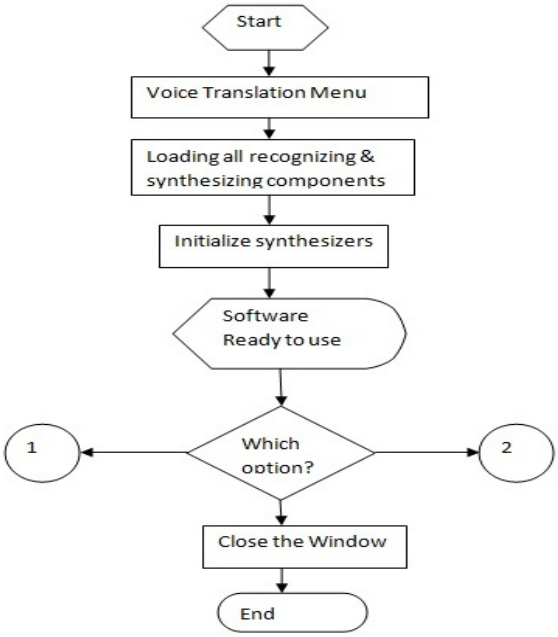

Figure 9: Flow chart for Voice Translation menu

In this diagram, the process starts with process voice translation 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 translation or Hindi to English translation of language or close the window. If user selects English to Hindi translation the control goes to connector named as 1.If user selects Hindi to English

7. Results and Discussion

1. Voice (In English) Recognition at Client Side

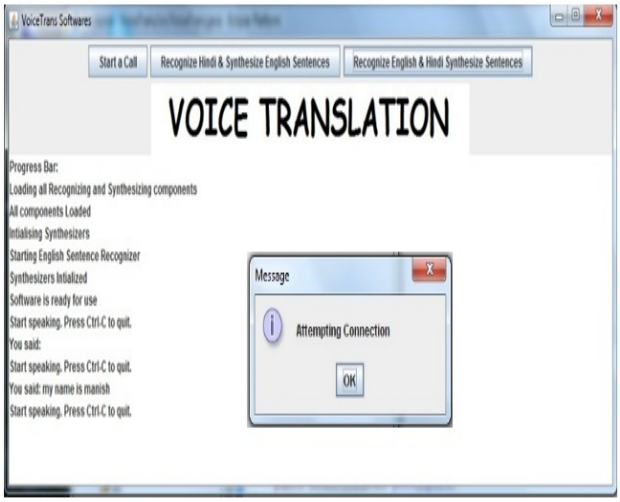

Figure 10: Voice (in English) recognition at client side

2. Output Voice (In Hindi) At Server Side

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

3. Voice (In Hindi) Recognition

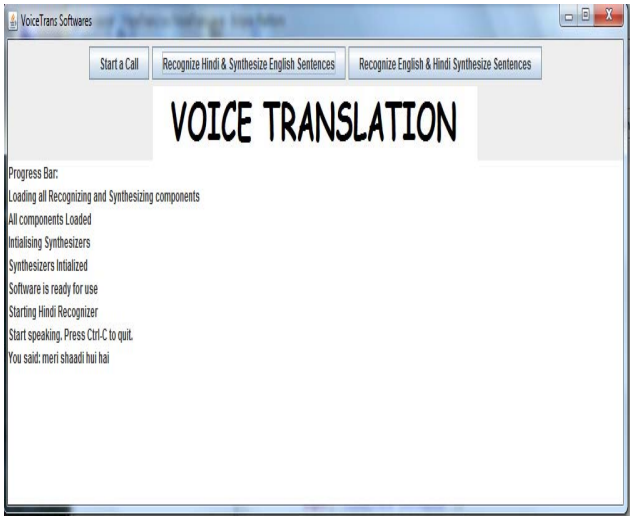

Figure 12: Voice (in Hindi) recognition

Voice (In English) Output

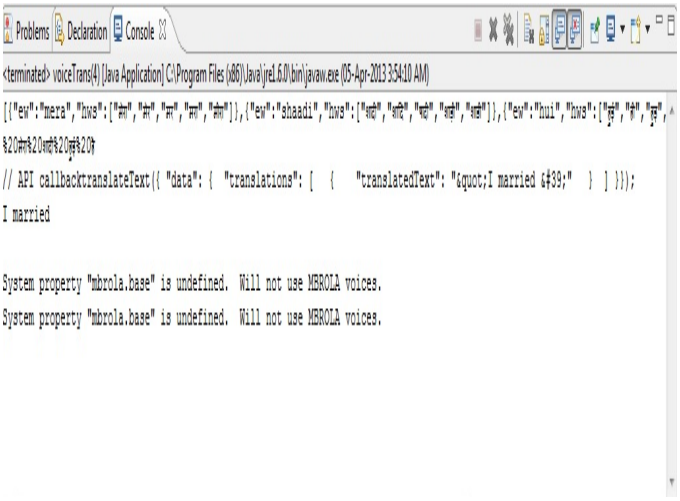

Figure 13: Voice (in English) output

8. Conclusion & Future Work

8.1 Conclusion

Language has always been a barrier to effective communication. As businesses expand and technology engulfs the entire globe, reliable and real-time translation 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 real-time communication between two users who do not speak a common language.

8.2 Future Work

The user will be given option to select the source and the destination language. The different types of languages will be added at both sides.

References

[1] Yen Chun Lin *"An optimized approach to voice translation on mobile phones", 2010.*
[2] Titus Flex FORTUNA" *Dynamix Programming Algorithms in Speech recognition", 2008,pp 94-99.*
[3] Srinivas Banglore, Vivek Kumar Rangarajan Sridhar, Prakash KolanLadan Golipur, Aura Jimenez"*Real-time Incremental Speech-to-Speech Translation 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 study", August 2002*