Language Recognition from Speech Using Machine Learning And Audio Translation

Summayya Banu[1], *Dr. K. Prasanna Lakshmi*[2]

[1]Student, Department of IT, GRIET, Hyderabad, TS, India.
[2] Head of Department, GRIET, Hyderabad, TS, India.

Abstract:

Language identification is used to determine the language being spoken in an audio when compared against a list of provided languages. This research paper displays the application and implementation of machine learning in speech recognition and translation. The background of creating a tool to identify the language of the real time audio, and how the process of translation is conducted, will be displayed. It will provide information about the method of speech recognition to get the desired results. With the implementation of machine learning, we will help in the real world to prepare automated voice translation without having to manually select the language or even when we are unaware of the language being spoken. We will be displaying the process of creating an application to trigger the translator and then using the operating system sound player, translated audio will be played. Our goal is to simplify the audio detection and translation process by creating a user-friendly interface so that it can be implemented by anyone.

Index terms: *Translator, Language, machine translation, speech-to-text, Google Translator, translator application*

DOI Number: 10.14704/NQ.2022.20.11.NQ66650 NeuroQuantology2022;20(11): 6524-6533

1. Introduction

An average person speaks approximately eleven thousand to twenty-five thousand words every day making speech the easiest way to express ourselves. No matter if it is a conversation, a dialogue, or a speech, or presentations or general day to day talks, we make use of speech to make others and as well as our own selves understand the actions and thoughts. If anyone in a conversation is ignorant of the language being spoken, the purpose of the conversation is not achieved [1]. Therefore, we require a tool that can connect such language barriers. Speech-to-speech Language translation is a system that can play a predominant role by ease of communication between people speaking in many languages [8]. Around the world, people are working to achieve this goal and implement it in a way that ordinary people can use.

In an age of globalization and endless economies, information exchange has become a necessary part of life, and communication is the most common and most effective form of connecting with each other. Globalization increases the need for communication between native speakers of different languages [4]. In fact, one of the biggest challenges facing information technology is overcoming barrier to communication in the global community and allowing them to express their feelings [29].

The objective of this experiment is to explore the field of automated speech translation to English, in particular. And in this modern world, the languages are used differently in different parts of the world. The only language which is globally recognized and used popularly is English. To automate the process of translation, we will design an interface to detect the audio from speech and convert it into English Language.

1.1 Existing System

Only a few award-winning literary works, press articles, and crucial professional documents are manually translated [3]. In our time, there were many complications in speech recognition. Latest tools and technologies helped engineers create several voice assistants, modules which are designed to work with voice and speech commands [30]. Few Gadgets are also planned to work with processed text such as Classifiers and also translators.

It is still necessary to manually select the language in all translators, regardless of readymade solutions from well-known world companies.

1.2 Proposed System

With the use of machine learning, we will help in the real world to prepare automated voice translation without having to manually select the language or even when we are unaware of the language being spoken. We will be writing a program to make the computer recognize the live speech input. It will convert the detected speech into text. After that, it will complete the task of automatically identifying the language of the speech input. Once done, it will also translate that given input to English and then that translated text will be read out by the computer's default audio player.

We will be creating a user friendly application start the program and then using the operating system sound player to play the translated audio. Our objective is to simplify the audio detection and translation process by creating a user-friendly interface so that it can be implemented by anyone, for any language.

2. Literature Survey

Audrey was the first speech recognition system to recognize only spoken numbers, according to research conducted [12] in 1952 at Bell Laboratories. It was ten years later, in 1962, that IBM developed another system that recognized 16 English words. Japanese researchers have developed hardware that can detect and analyze 4-word and 9-letter patterns in collaboration with the Soviet Union, the United States, and England

"Harpy" speech-processing software developed by Carnegie Mellon in 1971 and 1976 identified 1,011 words. For the first time, Threshold Technology and Bell Labs have been able to synthesize words to translate the speech of people from different languages. It was introduced in 1980 that the Hidden Markov model (HMM) could match up to 100 words and a few thousand words. [12].

Through the Microsoft Windows PC, Dragon Dictate had brought speech recognition to the workplace by 1990. In 1990s, the trend for speech detection in the workplace had continued. Apple also released a tool called ‘Speakable Items’ in the year 1993, an integrated and controllable software for its computers. Another major continuous speech recognition system was introduced in 1993 with Sphinx-II. As IBM's first commercial product capable of consistently recognizing speech, it released a tool called 'MedSpeak' a few years later [12].

In an interview held in 1997, Bill gates said, “In 10 years I believe that we’ll not only be using the keyboard and the mouse to interact, but during that time we will have perfected speech recognition and speech output well enough that those will become a standard part of the interface.” [18].

230 billion words of actual English words are integrated into Google's voice search. Google introduced voice recognition in 2015 and experimented with connectionist temporal classification (LSTM) (CTC) in Google Voice. [14].

There are also several practices and cases regarding the translation from one language to the other. If translation is an art, it is not easy [30]. In the thirteenth century, Roger Bacon wrote that the translation is correct, that the translator must know languages as well as knowledge of the translation; and after several interpreters did that, he wanted to accomplish the translation process without the need of translators [30]. As a researcher for the Rockefeller Foundation during World War II, Warren Weaver proposed [26] machine translation based on data theory and decoding. Years later, machine transfer research began in many American universities.

In 1994 the Language Machine Translation (LMT) programs were named as IBM’s computer based Translation Manager modules [18]. The interlingua pathway grows vigorously, even at the end of the DLT and Rosetta experiments. Carnegie Mellon University developed several models over the years based on its knowledge-based approach (Nirenburg et al. 1992). The company announced in 1992 that it would begin a collaborative project with Caterpillar to develop an effective CATALYST multilingual translation of technical manuals in the field of heavy earth moving equipment. The program includes a knowledge-based approach and controlled inputs.

Soon afterwards, in 1995, Babylon Fish discovered Altavista - a system that could automatically translate text into many languages [27]. The program was available online for free and transportation was provided to the public. These software transfer programs were used. This method eliminates the source material in the previous general translations.

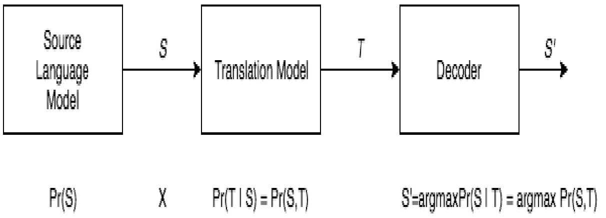
While there are only "fully automatic, with great free transfer capabilities", there are many programs today that can provide tangible results with strict limits. Many of these programs are available online, such as Google Translate and SYSTRAN, which make AltaVista's BabelFish (now Yahoo's Babelfish as of May 9, 2008) [29].

One of the most successful tools is Google Translate, a translator developed by Google in April 2006 [6]. It translates texts and media ,words, phrases etc including web pages.

Google Translate was originally intended to be a mathematical translation service [11]. Before it could be translated into the chosen language, the added text had to be translated into English first. SMT has grammatical precision since it utilises predictive algorithms to translate text. Aside from that, because to the ever-changing nature of language, Google did not initially hire experts to resolve this issue.

Brownet al. proposed using mathematical principles in machine translation in 1990 [2]. They developed a translation method in which the source text was divided into a series of fixed paragraphs, a dictionary was used to select a series of consistent paragraphs, and then the target language terms were renamed to construct the target sentence. They succeeded in developing mathematical strategies to automatically create word lists and arrange target word order, but they did not provide examples of translated sentences.

Fig. 1. Statistical Machine Translation



Bronet's research [20] continued and in 1993, she developed a set of five models that included the number of pairs of bilingual sentences as a metric for measuring the parameters of these models. The IBM alignment model was later renamed after them. Explain how bilingual sentences can be compared word-for-word.
Give an algorithm of real opportunities for each game in each pair of sentences. Although his research was limited to small translations into English and French, his word-for-word relationships within a sentence pair had improved significantly [20].

In 1996, Vogel Et. [21] The Markov model of the first hidden order was used by Al to develop a new word alignment model for mathematical machine translation because it resolved the time problem of speech recognition. Essentially, the goal was to give each word the chance to be justified in light of forgiving circumstances, not perfect conditions.The HMM-based model produced translation opportunities compared to the hybrid alignment model, and the local alignment was much smoother than the HMM-based model [21].

Hand. Al in 1999 [7], explained how to determine the categories of bilingual words that will be used in translating a mathematical machine. They developed a condition of preparation based on high probability estimation and explained the integration algorithm in more detail. Using bilingual pronouns improved mathematical translation considerably, based on their assessment.

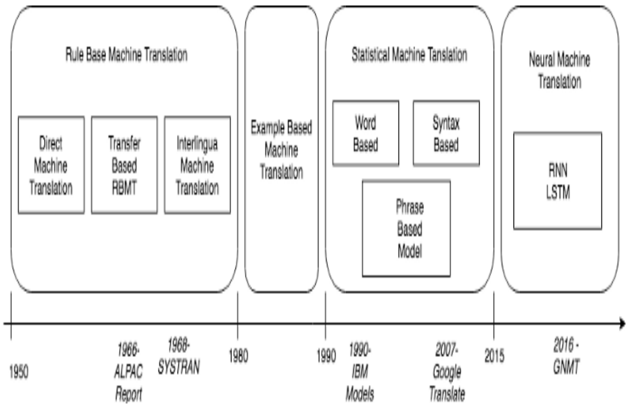
Yamada Et. Al (2001) suggested mathematical translation model based on the syntax [24]. He used stochastic actions in each field to transform the source language analysis tree into a unit of target language characters. A variety of linguistic differences were included in these activities, such as word order and punctuation. The model introduced a correspondence of the main names produced by IBM Model 5 [24].

A translation model based on new sentences and a recording algorithm proposed by Koehn et. Al (2003) [24] allowed them to evaluate and compare a set of translation models based on the sentences suggested previously.Koehn et al. (2003) suggested a model that was based on Brown et. al (1993) also used Bayes law to provide an opportunity to translate foreign sentence from French to English..

Chiang Et. Al [9] proposed a sentence-based machine translation model employing high-level sentences later in 2005: sentences containing clauses. His model was based on a bold grammar without context. The template creates a partial translation using hierarchical sentences, then compiles it sequentially with a standard sentence-based template. Rather than using the standard audio channel method, use a standard log-linear model.

With the advancement of machine translation process, statistical machine translation was then taken over by Neural Machine Translation [5].

Fig. 2. Evolution of Machine Translation


Rule Base Machine Translation
Direct Machine Translation
Transfer Based RBMT
Interlingua Machine Translation
Example Based Machine Translation
Statistical Machine Tanslation
Word Based
Syntax Based
Phrase Based Model
Neural Machine Translation
RNN LSTM
1950
1966-ALPAC Report
1968-SYSTRAN
1980
1990
1990-IBM Models
2007-Google Translate
2015
2016 - GNMT

According to the research paper by Dzmitry, Bahdanau, Kyunghyun Cho, Bart van Merriebboer in 2013 [5], Kalchbrenner & Blunsom proposed a new encoder-decoder-decoder-machine. They were based on the continuous express of sentences, words and phrases rather than on the meaning of individual translated words. This research was continued by Ilya Sutskever in 2014 and recommended the use of neural networks in an in-depth series of machine learning. With this method, input vector sequences of a fixed size are mapped using a long-term serial memory (LSTM), and then targeted sequences for the rider are compiled using some LSTM depth. Their results have shown that neural systems with broader translation, LSTM with higher vocabulary limits can do better than a standard SMT-based system.

In the year 2016,Mike Schuster, Zhifeng Chen, Yonghui Wu, Quoc V.LE and Mohammed Norouzi all together performed a research [12] and developed a model of common sequence-to-sequence learning framework by collaborating the machine translation work done by Sutskever et al.(2014) and Bahdanau (2014). In this experiment [12], they introduced Google's Neural Machine Translation (GNMTS) system that tries to solve any problems reported by other researchers, such as computational costs and longer training times. Their model consisted of a deep LSTM network consisting of eight encoders layers and eight decoder that used leftover communication and attention-grabbing connections from the output network to the transmitter. Their aim was to improve similarity and thus reduce learning time. For this, they connected the lower layer decoder and the upper layer decoder. They used speed and minimal arithmetic accuracy when performing sequential calculations that accelerated the final translation [12].

An extensive research by Benjamin, Martin (2019) [13] was also conducted by one of the researchers where a comparison of different language translation and its accuracies were conducted. In this experiment, 20 common English phrases were translated to 102 languages and then sent a re-worded explanation of the phrase to independent evaluators. For example, if we showed “out of steam” along with its explanation, some evaluators would see that a word for “steam” was given and therefore judge the translation highly, and give the explanation as “no more energy (exhausted)”, they were able to judge whether the proposed translation captured the usual English meaning [13]. The least recognized phrase, “out cold”, was only understandable to some extent 10 times, while the phrase “out of office” produced an understandable result of 83% of the time.

3. Methodology

There are various tools and packages involved that club together to produce our desired results. This software of Speech Recognition actually separates the recorded speech into individual sounds, analyzing each and every sound, using algorithms to determine the most likely word in the language, and duplicating those sounds into text [23].

This software of Speech Recognition uses Natural Language Processing (NLP) and Deep Learning Neural Networks. “NLP is a way for computers to analyze, understand, and derive meaning from human language in a smart and useful way,” as per ‘Algorithmia blog’ [23]. This software breaks down the given speech into fragments, then its translated into digital format to analyze the content pieces.

Now, this software makes decisions based on program and patterns of speech, creating ideas to find out what the user wants to say. After making decision of what users want to convey, then this software again converts this conversation into text.

In this speech recognition, the transcription might not be 100% accurate (some words, names, and details might be mis transcribed), but the user still understands the overall idea of the chunk of speech that was transcribed.

The way speaker pronounce clearly or speaks mumble , the speaking speeds, and even change in their voice volume can cause speech recognition to malfunction [3].

However, many modern speech recognition techniques uses Machine Learning Platforms(MLP). Hence, the software identifies the user’s speech patterns and variations and adjust it if the user continues to use the technology.

The methods involving the translation process also plays a major role and falls inline to be able to work on the output produced by the speech recognizer. The prominent tool that we take advantage of is the Google Translator (GT). Google Translator does not use Grammar rules as the algorithms as its algorithms are based on Pattern Analysis and Statistics and not on standard rules analysis.[14]. According to an article written by Franz Och [10], a Research Scientist, several research programs were provided with billions of text words, both Interactive Text (which includes human translation between languages as an example) and monolingual text in the target language. Mathematical learning methods were then used to construct the translation model. This has proven to achieve excellent results in research evaluations.

Underlining various approaches, research by Melvin Johnson, Fadi Biadsy, Wolfgang Macherey, Ye Jia, Ron J. Weiss, and Chen, Yonghui Wu [16], is designed for Spanish as a source and English as the target language. The technical feature in terms of raw code has not been available to the public yet. Sequence to Sequence Neural Network i.e., attention based is used in this Model. The entire structure of encoding and decoding is trained to make a speech spectrogram map directly to target spectrograms using speech pairs called utterances. One of the key features of this model is the power of the model to replicate the speaker's voice in the translated language [16].

Another similar translation approach was followed for translation of Indonesian language to English [28] where unsupervised and weakly supervised learning techniques were implemented that do not require large data sets, with annotations, to resolve the limitation of annotated corpora. In order to create the stochastic language model, it is appropriate to use annotation data when available and to use a supervised learning method to quantify the model parameter. In this case, an annotated corpus is created for many types of documents[17].

3.1 Working of Google Neural Machine Translation

The implementation of Google Neural machine translation, the preprocessor tokenizes sentences into words, then breaks input text into phonemes and conveys emotions [11]. The Source Sentence is converted into list of vectors, each vector per input mark. Now, the work of decoder is to generates each symbol at a time until and unless a special end of Sentence (SOS) is detected.

An research was conducted by Mohammad Norouzi, Quoc V. Le Zhifeng Chen, Mike Schuster and Yonghui Wu [11] According to this research, both encoder and decoder are being attached using attention module. This Decoder and Encoder allows the decoder to aim different regions of source sentence when its recording

This architecture was represented as follows:

Let (X, Y) be the source and pairs of targeted sentences. Where X = x1, x2, x3, ... xM be the order of the M symbols in the source sentence and Y = y1, y2, y3, ..., yN be the sequence of N symbols in the target sentence [11].

The encoder is a function of the following form:

$$\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M = EncoderRNN(x_1, x_2, x_3, ..., x_M) \tag{1}$$

In this equation, x1, x2, ……. xM contains vectors list which is of same (fixed) [11].

The quantity of both members in the list and also symbols are same in actual sentence (Here, M is the example). According to the chain rule the conditional probability of the pattern P(Y |X) can be computed as:

$$P(Y|X) = P(Y|\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_M) \\ = \prod_{i=1}^{N} P(y_i|y_0, y_1, y_2, \ldots, y_{i-1}; \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_M) \quad (2)$$

Here, y0 is referred as “beginning of sentence” symbol that is added at the start of each target sentence.

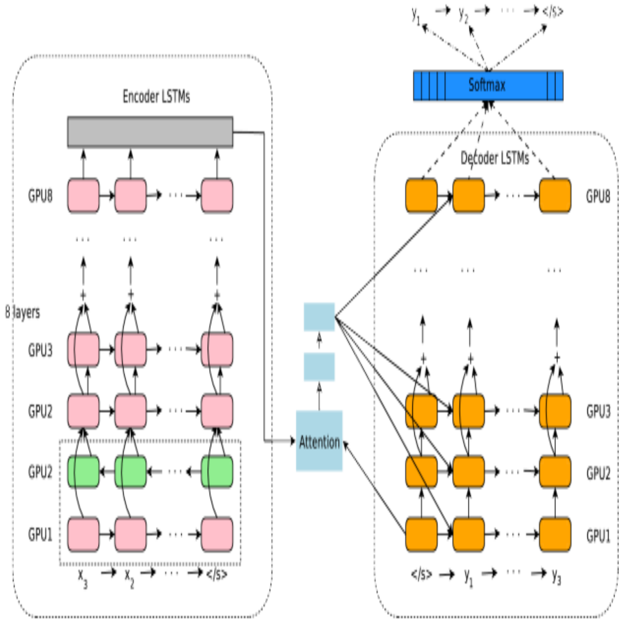
While reasoning we compute the probability of both the next symbol from Encoded Source Sentence and
the Decoded Target Sequence as:

$$P(y_i|y_0, y_1, y_2, y_3, \ldots, y_{i-1}; \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_M) \quad (3)$$

The decoder is carried out as a fusion of a Recurrent Neural Network and a softmax layer [11].

To forecast the next symbol, decoders of RNN produce a hidden state yi, which is passed to the softmaz layer.This produces a probability distribution over the candiate output symbols.

Fig 3. Model architecture of GNMT



The above image is Architecture image. The left one in the picture is Encoder network, to the right we have a decoder network and the attention module. The lower layer of the encoder is bi-directional: the pink nodes collects data from left to right and then the green nodes collect info from right to left [11]. The other one i.e, encoder layers are uni-directional.

The residual connections starts from the 3rd layer from the lowest, inside the encoder and decoder. To speed up the training process, this model is split into multiple GPUs.
Here in this setup, there are eight 8 LSTM endoder layers (Seven Uni-Directional and 1 bi-directional layer), also there are 8 decoder layers [11].Now, with the help of this setting, a single model is divided into 8 modes and placed on 8 different GPUs for a single hosting machine.

During training, the layers of the lower bi-directional encoder are calculated equally first. Once both are completed, layers of single-directed decoder can start to build a computer, each with a different GPU [11].

To provide as much consistency as possible while using decoder layers, the lower decoder layer output was used only to obtain a reciprocal attention setting, which transmits directly to all remaining decoder layers.

This Softmax is a layer which is also split and placed on multiple GPUs. Now same GPU was used to drive the encoder and decoder network, or they were run on a different set of GPUs which are dedicated separately, depending on the output vocabulary size [11]. Below is the formula for the reference a_i for the current time step is

$$s_t = AttentionFunction(\mathbf{y}_{i-1}, \mathbf{x}_t) \quad \forall t, \quad 1 \le t \le M \\ p_t = \exp(s_t) / \sum_{t=1}^{M} \exp(s_t) \quad \forall t, \quad 1 \le t \le M \quad (4) \\ \mathbf{a}_i = \sum_{t=1}^{M} p_t.\mathbf{x}_t$$

where Attention Function is a feed forward network in this model which has one hidden layer.

Inspired by modeling differences between the outputs and targets of an intermediate layer, which has worked well for many demonstration in the past, residual connections between Long Short Term Memory layers in the stack were introduced.

Precisely, let $LSTM_i$ and $LSTM_{i+1}$ be the i-th and (i + 1)-th LSTM layers in a stack, where in the parameters are Wi and Wi+1 respectively. At the t-th time step, for the stacked LSTM without residual connections [11], we get:

$$\mathbf{c}_t^i, \mathbf{m}_t^i = \mathrm{LSTM}_i(\mathbf{c}_{t-1}^i, \mathbf{m}_{t-1}^i, \mathbf{x}_t^{i-1}; \mathbf{W}^i) \\ \mathbf{x}_t^i = \mathbf{m}_t^i \quad (5) \\ \mathbf{c}_t^{i+1}, \mathbf{m}_t^{i+1} = \mathrm{LSTM}_{i+1}(\mathbf{c}_{t-1}^{i+1}, \mathbf{m}_{t-1}^{i+1}, \mathbf{x}_t^i; \mathbf{W}^{i+1})$$

where x_t^1 is the input to $LSTM_i$ at time step t, and m_t^1

and $c_t^1$are the hidden states and memory states of $LSTM_i$ at time step t, respectively.

Residual Connections among $LSTM_i$ and $LSTM_{i+1}$ can be written as per the above equations are:

$$c_t^i, m_t^i = LSTM_i(c_{t-1}^i, m_{t-1}^i, x_t^{i-1}; W^i)$$

$$x_t^i = m_t^i + x_t^{i-1} \tag{6}$$

$$c_t^{i+1}, m_t^{i+1} = LSTM_{i+1}(c_{t-1}^{i+1}, m_{t-1}^{i+1}, x_t^i; W^{i+1})$$

The gradient flow is greatly improved by residual connections, that makes to train deep decoder networks with encoders.

4. Implementation

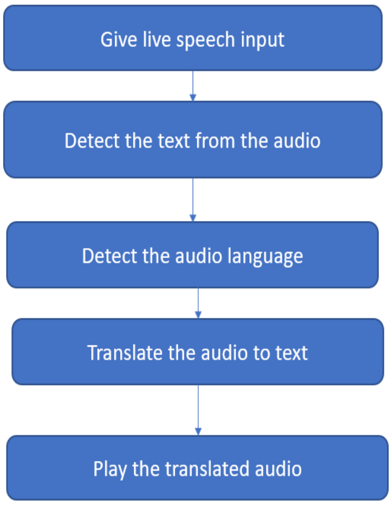
We make use of several packages available in python such as Speech Recognition, GoogleTrans, Natural language processing (NLP) and Google Text-to-Speech (gTTS).

The main objective of using python in our research experiment is because it is Object Oriented, easy to code, a fun programming language. The number of lines or codes that we enter in this language to perform a function is far lesser in Python, in comparison with other languages [15].

These packages provide us with plenty of predefined functions that help us construct all the related modules. Every package plays a major role in each step of the application creation.

A simplified overview of the complete process can be represented as follows:

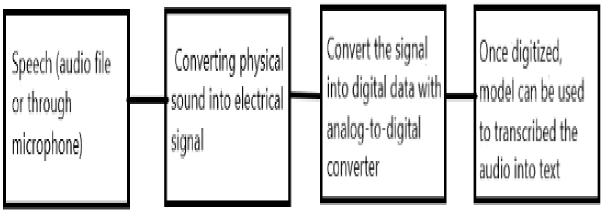
Fig 4: Implementation overview


Give live speech input
Detect the text from the audio
Detect the audio language
Translate the audio to text
Play the translated audio

Initially, we need to understand the process of taking the input speech and how to identify meaningful words. This process will be implemented using the speech recognition module. In human computer interaction, this Speech recognition is considered the vital tasks. This task of detecting the human speech is accomplished by this package. We will be using speech recognizer to recognize the audio and convert it into text. To perform this operation, we will be using the python inbuilt library SpeechRecognition.

The working of a speech recognition module can be shown as follows:

Fig 5: Speech Recognition process


Speech (audio file or through microphone)
Converting physical sound into electrical signal
Convert the signal into digital data with analog-to-digital converter
Once digitized, model can be used to transcribed the audio into text

After recognizing the text from our audio file, we will process the text to create meaningful sentences. To perform this operation, we will be using the Natural language Toolkit.

After the audio is recognized and converted into text in the first stage, the text needs to be translated. Here, we design the program in such a way that it automatically recognizes the language being spoken in the input audio.

The Google Translator (GT) can get it for free of cost and considered asynchronous. Python library is the one which implements the GT API. It uses the Google Translate Ajax API to call methods like detect and translate.

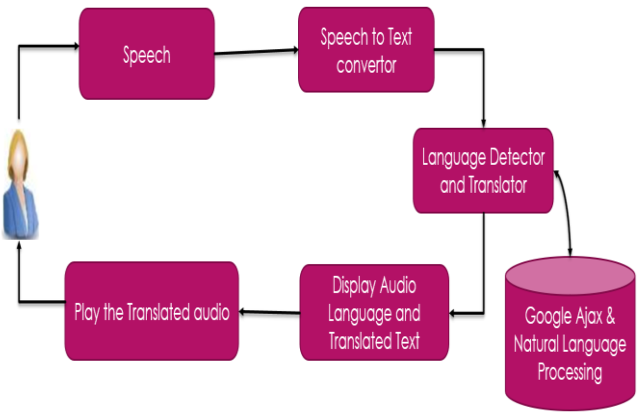
The workflow of Google Translator Toolkit can be described [31] as follows.

1. First, users gives and audio input that they want to translate.
2. Next, it searches for all available translation sites for previous translations of each section. If there is a previous human translation of the segment, Google Translator Toolkit selects the advanced search result and 'pretranslates' the part of that translation.
3. If none of the available previous human translation matches, machine translation is used to produce 'automatic translation' of the part, without any involvement from the human translators.

After the text is translated, we need our program to read the text. This can be accomplished by using the Google Text-To-Speech package. An Python library called GTTS (Google Text-to-Speech) interfaces with API of Google Translator’s text-to-speech. After the audio output is generated, we trigger the default audio system to play the translated audio, resulting in end-to-end speech detection and translation.

The architecture of our research experiment can be shown as follows:

Fig 6: Implementation Architecture


Speech
Speech to Text convertor
Language Detector and Translator
Google Ajax & Natural Language Processing
Display Audio Language and Translated Text
Play the Translated audio

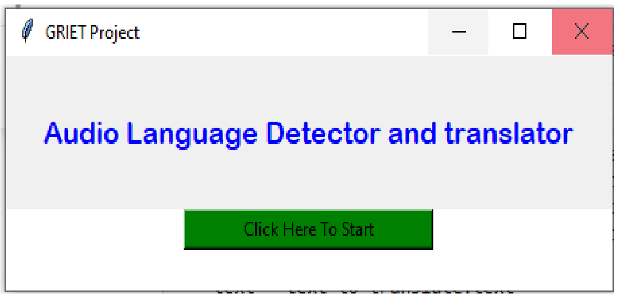
As we aim to simplify the complete process so that it can be implemented by anyone, we create a graphical user interface, a front-end application for the users to directly interact with the program. This operation is carried out by the Tkinter application package [22]. This package is easily available for most Unix platforms, including macOS and Windows systems. Tkinter is transparent and easy to use when compared to other Graphical user interface frameworks. These advantages of Tkinter app makes to choose Tkinter app to build Graphical User Interface GUI applications in Python and in building something that's functional and cross-platform easily [22].

An application that we create will be linked internally with the program so that any user, even who does not have any knowledge about programming or how to execute a program, can also get the desired results.

5. Results Analysis

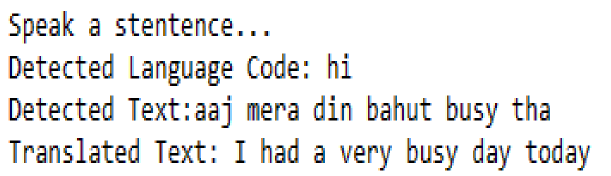
The results of the described experiment will be explained in this section After completing our code, we need to execute our program to get our desired output. In our program, a Tkinter application pops as our user interface. This becomes the medium for the users to interact with the program and start the experiment.

Fig 7: Front End Output



After we click on the Button 'Click Here to start', it will invoke our translator function and start the program. Once the function is triggered, the system microphone is triggered to record live audio. Using the detected audio, the language of the speech is detected, detected text is displayed, along with the English Translated Text.

Fig 8: Output Screen



```
Speak a stentence...
Detected Language Code: hi
Detected Text:aaj mera din bahut busy tha
Translated Text: I had a very busy day today
```

As we can see here, the audio language is automatically detected and then translated to the English language.

Also, clicking on the button again will start the execution again without having to run the entire program again. While the written text is shown in the console, the translated audio is played using the computer's audio player.

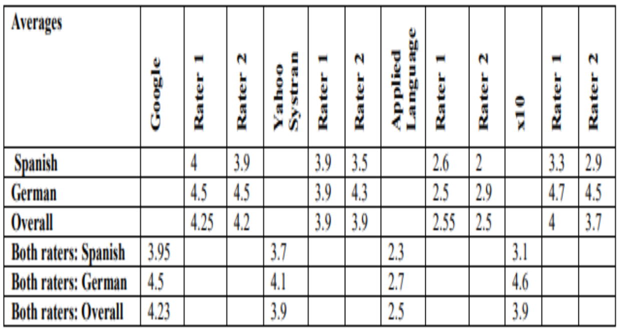
Fig 9: Output Audio

The output audio can also be replayed as many times as required.

Based on the comparison of human translation, we show that our Google Neural Machine Translator system performs in a manner comparable to standard bilingual human translators [11]. On several popular language pair pairs, this GNMT system delivers a roughly 60% reduction in translation errors over the previous phrase-based production system..

In one of the research performed by JohnWee, Mahesh Vanjani, Milan Aiken and Kaushik Ghosh,[25], a comparison was done between different available translators in different languages. In this research, a study was done for the languages Spanish and German by 2 interpreters to compare the tools Google Translator, Applied Language, x10 and YahooSystran.

The interpreters rated the machine translations from 0-5 ratings for several machine translated paragraphs and their average was determined to identify which tool is performing the best.

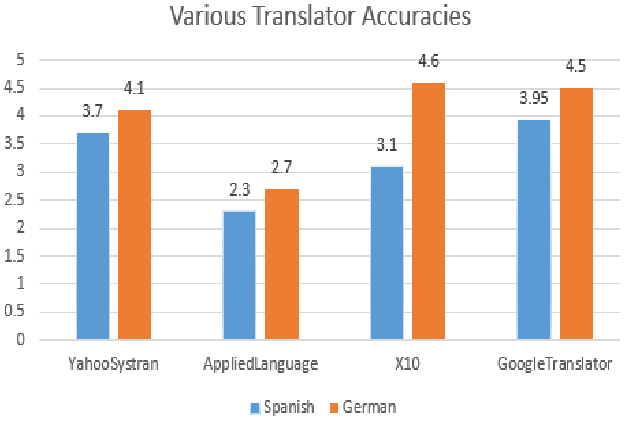


Averages	Google	Rater 1	Rater 2	Yahoo Systran	Rater 1	Rater 2	Applied Language	Rater 1	Rater 2	x10	Rater 1	Rater 2
Spanish		4	3.9		3.9	3.5		2.6	2		3.3	2.9
German		4.5	4.5		3.9	4.3		2.5	2.9		4.7	4.5
Overall		4.25	4.2		3.9	3.9		2.55	2.5		4	3.7
Both raters: Spanish	3.95			3.7			2.3			3.1		
Both raters: German	4.5			4.1			2.7			4.6		
Both raters: Overall	4.23			3.9			2.5			3.9		

In this research [25], each interpreter checked the machine translated sentence from Spanish to English and rated its accuracy. Similar, sentences translated from German to English were also rated by both the interpreters. After that, an overall score was determined by calculating the average of the individual scores.

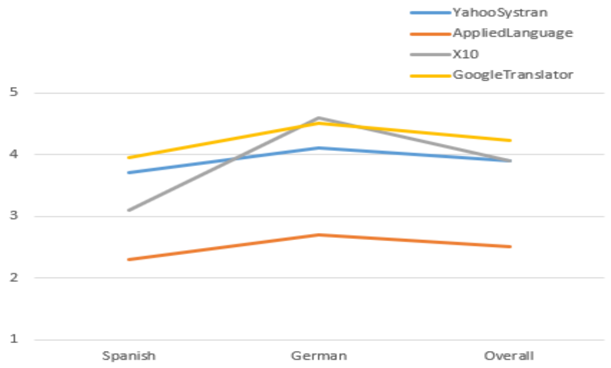
As per the human ratings, Spanish was best translated by the Google Translator and German was best translated by X10 translator [25].

Fig 10: Bar Graph for Various Translator Accuracies



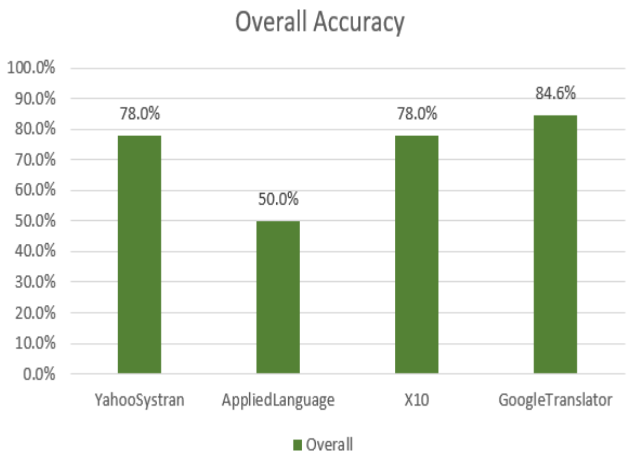
The line graph to contrast the language versus the translator tool can be shown as:

Fig 11: Line Graph for Translator performance



As seen in the above graphs, Google Translator performed the best overall with 84.6% accuracy score, whereas Yahoo Systran translator and X10 translator had an equal overall score of 78%. Applied Language translator service stood in the last position. Percentagewise bar graph with 50.0% accuracy is depicted in the Overall accuracy chart for each translator.

Fig 12: Overall Accuracy Bar Graph



6. Conclusion

A Speech Recognition program is created that converts the live speech input given into text format, allowing the machine to process the human input. Using the text produced, our program automatically detected the language of the source input and then converts it into English. After that, the translated text is again, converted into speech. Our finally produced translated speech audio is then played through the computer speakers.

Few Advantages of using this translator are:

- Users can use this application to easily translate the audio from their native language to English for better communication.
- Speech recognition allows the elderly and visually impaired users to communicate with products and services quickly.
- They can also use the application to learn the language as well, by using the replay translated audio feature.

This is a complete end – to – end application which does not require any human intervention throughout the process.

We can also further extend our research experiment by adding more features for audio detection, emotion recognition and translation.

References

[1] Harvard Business Review. 2022. *The Power of Talk: Who Gets Heard and Why*.

[2] Della Pietra, Vincent J. "Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin." *Readings in Machine Translation*: 355.

[3] "Evolution Of Speech Recognition Technology - From Audrey To Siri". *Verbit*, 2019,

[4] Limbu, Sireesh Haang. "Direct Speech to Speech Translation Using Machine Learning." (2020).

[5] Cho, Kyunghyun, et al. "On the properties of neural machine translation: Encoder-decoder approaches." *arXiv preprint arXiv:1409.1259* (2014).

[6] Analytics India Magazine. 2021. *Why Speech-to-Speech Translation Is So Important for Google*. [online] Available at: <https://analyticsindiamag.com/why-speech-to-speech-translation-is-so-important-for-google/> [Published 7 October 2021].

[7] Och, Franz Josef, Christoph Tillmann, and Hermann Ney. "Improved alignment models for statistical machine translation." *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. 1999.

[8] Arora, Karunesh, Sunita Arora, and Mukund Kumar Roy. "Speech to speech translation: a communication boon." *CSI transactions on ICT* 1.3 (2013): 207-213.

[9] Chiang, David. "A hierarchical phrase-based model for statistical machine translation." *Proceedings of the 43rd annual meeting of the association for computational linguistics (acl'05)*. 2005.

[10] Koehn, Philipp. *Statistical machine translation*. Cambridge University Press, 2009.

[11] Wu, Yonghui, et al. "Google's neural machine translation system: Bridging the gap between human and machine translation." *arXiv preprint arXiv:1609.08144* (2016).

[12] "The History Of Voice Recognition Technology - Blog". *CONDECO GROUP LTD*, 2018,

[13] Benjamin, Martin (2019). "Evaluation Scores of Google Translate in 102 Languages". *Teach You Backwards*. Retrieved December 26, 2019.

[14] Turovsky, B. "Found in Translation: More Accurate, Fluent Sentences in Google Translate. 2016." (2017)

[15] "Why Python Is Essential For Data Analysis And Data Science". *Simplilearn*, 2021, https://www.simplilearn.com/why-python-is-essential-for-data-analysis-article

[16] Jia, Ye, et al. "Direct speech-to-speech translation with a sequence-to-sequence model." *arXiv preprint arXiv:1904.06037* (2019)

[17] Riza, Hammam, and Oskar Riandi. "Toward Asian speech translation system: Developing speech recognition and machine translation for Indonesian language." *Proceedings of the Workshop on Technologies and Corpora for Asia-Pacific Speech Translation (TCAST)*. 2008.

[18] "Evolution Of Machine Translation". *Medium*, 2019, https://towardsdatascience.com/evolution-of-machine-translation-5524f1c88b25.

[19] RACOMA, B., 2018. *The History of Translations (Past, Present and Future)*. Day Translations Blog.

[20] Brown, Peter F., et al. "The mathematics of statistical machine translation: Parameter estimation." *Computational linguistics* 19.2 (1993): 263-311.

[21] Vogel, Stephan, Hermann Ney, and Christoph Tillmann. "HMM-based word alignment in statistical translation." *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*. 1996.

[22] "Tkinter — Python Interface To Tcl/Tk — Python 3.10.1 Documentation". *Docs.Python.Org*, 2021, https://docs.python.org/3/library/tkinter.html

[23] "What Is Natural Language Processing? Introduction To NLP". *Algorithmia Blog*, 2016, https://algorithmia.com/blog/introduction-natural-language-processing-nlp.

[24] Koehn, Philipp, Franz J. Och, and Daniel Marcu. *Statistical phrase-based translation*. UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY INFORMATION SCIENCES INST, 2003.

[25] Aiken, Milam, Kaushik Ghosh, John Wee, and Mahesh Vanjani. "An evaluation of the accuracy

of online translation systems." *Communications of the IIMA* 9, no. 4 (2009): 6.

[26] Hutchins, John. "The history of machine translation in a nutshell." *Retrieved December* 20.2009 (2005): 1-1.

[27] Hampshire, Stephen, and Carmen Porta Salvia. "Translation and the Internet: evaluating the quality of free online machine translators." *Quaderns: revista de traducció* (2010): 197-209.

[28] Utami, Silvia. "The source of errors in Indonesian-English translation." *Jurnal Kata: Penelitian tentang Ilmu Bahasa dan Sastra* 1.2 (2017): 192-202.

[29] Ahmad Abuarqoub, I. "Language barriers to effective communication." *Utopía y Praxis Latinoamericana* 24 (2019).

[30] Towards Data Science. 2021. *Building an Intelligent Voice Assistant from scratch*.

[31] Google Translate Blog. 2010. *Translating Wikipedia*. [online] Available at: <https://translate.googleblog.com/2010/07/translating-wikipedia.html>