COVID – 19: A focus on codemixing and codeswitching in Tamil speech to text

Abstract— This paper attempts to develop an application that converts Tamil language speech to Tamil text, with a view to encourage usage and indirectly ensure linguistic preservation of a classical language. The application converts spoken Tamil to text without auto-correction, code-mixing or code-switching. Tamil is a syllabic language, similar to other Indian languages and some unique features such as instances of allophones, short & long vowels and lack of aspirated stops produces some challenges in developing a speech to text app. This project is a technology demonstration of a complete web application, which, when perfected, could be used to act as a teaching tool to encourage correct pronunciation of syllables and words for native and non-native Tamil speakers. A report by the Business Standard India e-publication in the year 2019 highlighted the decline in the usage of the Tamil language, and indeed separate reports in Singapore and Malaysia, which both have large numbers of Tamil speakers indicate that there have been concerns about the relevance and usage of the Tamil language as a spoken means of communication among the community. The research maintains that it is important to maintain the utilization of Tamil language via technology to help preservation of one of the oldest surviving languages in the world. The work further emphasizes on the indigenous design considerations for such applications which may be different to traditional software engineering approaches.

Index Terms— Automatic speech recognition (ASR), Analog-to-digital converter (ADC), Application Programing Interface (API) Tamil, Speech to Text, web application, Software Engineering design

1 Introduction

There is a growing interest in speech to text applications as they can be used for a variety of functions in different areas such as education or business. These applications can be accessed on desktops, smart phones, or tablets to support the users and make their lives easier. However, there are different levels of ability and complexities among the speech to text applications which is based on the machine learning used by each application [1]. Among these applications are Dragon Professional, Speechmatics, Braina Pro, Microsoft Azure, and Dictation which supports Tamil Language. According to Tebelskis [2], speech is a natural means by which people use as a communication tool. In early childhood, without guidance, people normally learn all the relevant skills and continue to rely on speech communication throughout their lives. In fact, speech is regarded as one of the primary faculties of language, i.e. innate and biologically determined, which means that people start to communicate without the help of anyone [3].

Some of the human organs such as vocal tracks and articulators have features that are nonlinear – and are being influenced by factors such as human gender, educational achievement, and human emotions. These factors can affect human voice, accent, pronunciation, tone, and device's volume. Furthermore, during transmission, background noise and echoes may affect and cause distortion in the speech pattern. Also, electrical, and electronic features such as telephone and electronic devices may affect signal transmission from one node to another node. All the different individual effects can cause speech recognition to be more complex than speech generation.

It is widely noticed that people are comfortable with speech that many users prefer to communicate with computers using voice, rather than resorting to simplistic interfaces like keyboards and pointing tools. There are many applications that can benefit from the speech interface such as dictation tools and any spoken database querying. There have been substantial efforts since the 1950's to lead the work on automatic speech recognition. However, computers performing speech recognition is still not compared to the level of human performance and it is still an area that needs further research and innovations [2].

It is with no doubt that the human brain and the conventional computer work differently using different paradigms. Tebelskis [2] mentions that computers use a complex processor with specific instructions and local memory. On the other hand, the human brain uses parallel array of simple processing elements, i.e. neurons, bound by weight, i.e. synapses, which are changed with the individual experience to support the integration of multiple constraints. This paper extends on [30] and uses the conceptual framework proposed by[31].

1.1 Structure

The structure of the paper is as follows:

- Section 2 includes the literature review including the fundamentals of speech recognition and how the speech to text application work as well as its benefits and challenges.
- Sections 2, 3, 4, and 5 describe the methodology and the development of the speech to text system including the testing carried out on the developed application.
- Section 6 discusses the objectives, challenges and how these were dealt with on the project.
- The final section provides conclusion for the future research and development directions.

1.2 Aim

The aim of the project is to design and develop a prototype that converts Tamil speech into Tamil text without the effect of code-mixing and code switching.

1.3 Introduction to Tamil Language

Tamil is one of the Dravidian languages which is spoken mostly in the state of Tamil Nadu. There are also large Tamil-speaking populations elsewhere, including Sri Lanka, Malaysia, and Singapore, where Tamil has the status of a national language. According to Vistawide, there are over 68 million native speakers of Tamil. Tamil is a diglossic language which has two varieties that are used in different conditions [4]. The formal or ' literary ' type still largely consistent with Tamil grammar Pavanandi's standards set in the thirteenth century and it is used in almost all written media, as well as some high-register functions. Colloquial Tamil is used in all other contexts and is distinguished by major geographical and social variations [5].

As for speech to text applications in Tamil, there are currently several applications that support Tamil such as Azhagi Android App, Tamil Voice Typing and Tamil Voice to Text on Google's Play Store. These applications are widely used because they are convenient and quick. However, there are many concerns regarding the accuracy of these applications. The main reason for this lack of accuracy is that there is sometimes a nuance in the written sentence that a machine can't comprehend the same way as people [6].

2 Literature Review

2.1 Fundamentals of Speech Recognition

Speech recognition is a multi-level pattern recognition process that analyses and integrates acoustic signals into a hierarchy of subword units (e.g., phonemes), terms, phrases, and sentences [7]. That level can provide additional temporal restrictions, such as known word pronunciations or legal word sequences, which can compensate for lower level errors or uncertainties. A hierarchy of constraints can best be manipulated by probabilistically integrating decisions at all lower levels and making discrete decisions only at the highest level.

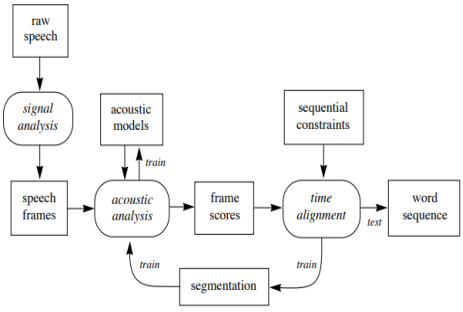


Figure.1: Structure of a standard speech recognition system [2]

Figure.1 demonstrates the framework of a typical speech recognition system. Usually, the raw speech is measured at a high frequency, e.g. 16 KHz (kilohertz) over a microphone or 8 KHz over a radio, resulting in a series of amplitude values over time. Initially, this raw speech should be transformed and compressed, to simplify subsequent processing. There are many techniques for analysing the signal which can extract useful features and compress the data by a factor of ten without losing important information. Of the most popular ones: Fourier analysis (FFT), Perceptual Linear Prediction (PLP), and Linear Predictive Coding (LPC) [2].

Speech is the basic, common and efficient form of communication method for people to interact with each other [8]. Currently, speech technologies are available for a limited but interesting variety of tasks such as personal assistants in smartphones, dictation and voice command in cars, as they enable machines to respond correctly to human speech and respond with useful but valuable services. In essence, executing commands on a system is faster using speech rather than using keyboards, so people may always prefer such a system. Communication among the human being is often characterised by spoken language, as such, it is natural for people to expect speech interfaces with a computer. This is where speech recognition systems come in as speech-to-text systems allow people to carry out certain tasks with a higher level of efficiency. Speech recognition system: speech-to-text is the

process of converting an acoustic signal which is captured using a microphone to a set of words [8]. The recorded data can be used for document preparation among many other uses.

The last 5-10 years in Automatic Speech Recognition (ASR) have focused primarily on minimising errors while decoding speech inputs. There is a need to make speech recognition available for more languages and cover broad topics. Speech recognition applications cannot always correctly convert spoken words. This is because machines understand contextual meaning of words and sentences are not on par with humans, creating misinterpretations of what the speaker meant to say or accomplish [9].Other scholars argue that developing a conversion system for Tamil involves many challenges including: (i) lack of standard, transcribed speech corpus, (ii) unlimited vocabulary problem, and (iii) lack of standardized lexicon. In spite of a rich heritage and literature, Tamil can be considered as low- resourced in this respect [10].

Despite these challenges, we propose to develop a speech to text web-based application that would allow users to speak and obtain text in the Tamil Orthography. The system will be consistent with the pronunciation of the user and conforms with the syntax of the target language thus providing the user with output directly equivalent to the words produced. This application will meet the requirements enabling native Tamil speakers to preserve the linguistic heritage.

2.2 How does the system work?

Many of the earlier speech to text conversion systems tried to apply a set of grammatical and syntactical rules to speech so if the words spoken fit into a certain set of rules, the program then could determine what the words were. However, accents and dialect can vastly change the way certain words or phrases are spoken so the rules-based systems were unsuccessful because they couldn't handle these problems.

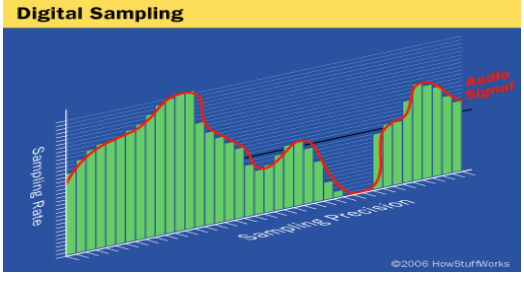


Figure.2: Speech to Data (HowStuffWorks, 2006)

An ADC - is a tool that converts analogue waves of voice into digital data by sampling the sound input. As the sampling and the precision rates of the sound increase, the output quality of the sound enhances.

There are two crucial elements that the user needs in order to use the speech to text software - including a working microphone that can pick speech and a working internet connection. Because smartphones are small and have limited space for software, much of the speech-to-text process is conducted on the server. When a user speaks into the microphone, the phone sends the bits of data of the spoken words created to the server, where the database is accessed by the software to match the spoken words to the best matching text. From the server it is returned to the client machine in form of some text file.

The program will examine phonemes in the context of the other phonemes around them and will compare them to a library of known words. The program then determines what the user was probably saying and outputs it as text

Figure.3: Speech to Text Online Tools (Dhani,2016)

2.3 Benefits of using Tamil Speech to Text applications

- **Ease of communication**

For native Tamil speakers, it will provide an opportunity to easily communicate with others via text message as it can be dictated by the user and converted to text to be sent to the receiver.

- **Linguistic preservation**

Can be used as a tool to encourage the use of the language as a medium of communication. It is therefore important to exclude features like code-mixing and code-switching and incorporate linguistic features to facilitate the process of linguistic preservation by the community.

- **Time saved with increased efficiency and less paperwork**

The average person can type between 38-40 Tamil words per minute [11]. When you replace this traditional method with a mobile transcription app to speak into, one is able to boost this speed by nearly 4 times, an average of 150 words per minutes using a speech to text app

- **Multitasking**

Dictation on the go - eliminating the need to perform dictation tasks on larger and more cumbersome devices such as laptops or personal computers

- **Accessibility**

Devices such as mobile phones, tablets and personal computers (PC)can be easily handled using the developed system.

2.4 Developmental Challenges

- Efficiency & Time

Although it is widely believed that computerising a process would accelerate it, speech recognition systems is in exception in this case. Using a voice app can take longer than using a traditional text-based version. The reason for this is the varied human voice patterns that speech interface is still learning to adapt to. Thus, users often must modify their pronunciation by slowing down or being more precise than usual [9].

- Different Accents

Speech interfaces are challenged when voice inputs divert too much from the usual pattern. In this case, the various accents of people may present a major challenge. Although systems are improving, there is still a big difference in their ability to understand for example American or Scottish English. It is not only by different accents, but even a sick voice cause by flu may lead to wrong interpretation of voice commands [9]. With regards to the Tamil language, although intonation and speech may vary according to geographical region when speaking day to day Tamil, formal Tamil speech does not recognize accents [3].

- Background Noise

It is always preferable to have a quiet environment to make the most of speech interface. It can be very challenging for the users if there is an excessive amount of background noise as speech recognition may not work efficiently outdoors, or in large public spaces. This problem could be partly solved by using specific microphones or headsets, but an additional device may be required which may not be desirable in many instances [9].

- Code Switching

It is when a speaker alters between two or more languages in a sentence or conversation. Example – Are we eating *chez ta mere demain*? (English + French)
A Tamil example – Book tableல இருக்கு

Code Mixing

Code mixing is the mixing of two languages to form a word, phrases or clauses. A glaringly obvious example would the title of this paper (கோ) *Co*vid-19! Some other examples of code-mixing are as follows: classஉக்கு, roadல

- Language maintenance

Many people may have difficulties speaking their native languages with correct pronunciation, due to having been brought up in a different geographical location or country to where their parents or even grandparents come from. For example, one of the team members was born in Ghana and at the age of three, he migrated to Italy. Due to this, he is able to speak fluent Italian however struggles with his native language from Ghana (Akan). Another member of the team was born in Malaysia, as were his parents. However, his grandparents were migrants from India and therefore the spoken variety of the Indian languages differ between the different generations in his family. However, one of the objectives of this project is to convert the exactly spoken speech into text form therey encouraging the migrants to retain original features of Tamil language.

2.6 Prototype

The proptotype was executed and tested by the following:

1. Download Netbeans IDE 8.2 RC
2. Download JDK 8u111 with NetBeans 8.2 RC
3. NetBeans Connector
4. Open Netbeans to run the project
5. Go to ‘File’ ‘Drop-down’ and click on ‘Open project’
6. Browse folder containing the project and click open
7. The folder will then be loaded to Netbeans
8. Select the project and click on ‘Run’ option on the menu
9. The application will automatically run on Google’s Chrome browser

2.7 Quality Expectations

Many people will use the application daily for a variety of use cases. Those people who use the application in these use cases must be happy with the quality of the systems they use, otherwise they will avoid using them [12]. To date, many firms and academic researchers developing and deploying applications continue to strive to improve the quality of the produced. This too is an obvious indication that the quality level is sub-optimal and can be enhanced [12]. The quality control the group is going to adopt is to perform product testing with the in-tention of identifying and fixing errors. Product was

done as a means of ensuring that the device complies with stated specifications and product validation as a means of demonstrating whether the software meets its intended use. If the device performance is good enough for some application areas, then efficiency must be measurable [12].

2.8 Considerations with regards to Tamil

- **Time and lack of efficiency**

A common belief is that computerising a process would accelerate it. Unfortunately, when it comes to voice recognition systems this may not always be the case. Using a voice app can take longer than using a traditional text-based version.
This is mostly because of the varied human voice patterns that VUIs are still learning to adapt to. Thus, users often have to modify their pronunciation by slowing down or being more precise than usual.

- **Accents and local differences**

Often, VUIs are challenged when voice inputs divert too much from the mean. Particularly the accents may pose a major challenge. Although systems are improving there is still a big difference in their ability to understand for example American or Scottish English. Even a simple cold may cause voice commands to be recognized incorrectly. As mentioned previously in section 2.4, formal spoken Tamil does not recognize accents, however tone and intonation of individual speakers can vary and therefore pose a challenge to VUI's.

2.9 Development of the prototype

This project was developed by a team of seven who have diverse experience and skillsets. A mixed approach was chosen to develop the software because is a method that is suitable for a small team.

The project was prototyped, evaluated and tested as the system progressed. The final prototype was executed by allowing a native Tamil Speaker to speak into the microphone of the system to verify the resulting text and complete the test case.

2.10 Feasibility

This section examines the feasibility of developing the speech to text application on a widespread basis, in the context of technical and economic feasibility.

With regards to the technical aspect of the application, Google's Cloud API was the main contributor to developing it, and is freely available to anyone wishing to develop a speech-to-text application. As other speech-to-text applications exist, code can be re-used to develop such an application should there be a need for it such as identified in this project.

There are a number of free website hosting sites that could be used; however, it is to be noted that many of them do not support Java based applications, and alternate methods of hosting need to be identified assuming that the developed speech-to-text app uses Java as the programming language.

The requirements for this project as well as various literature suggest there is a need for the promotion of usage for the Tamil language. The purpose of this is to create an awareness of the uniqueness and heritage of the language. Therefore, it appears that there is a real need for creating such an application, although the team concurs that further research, analysis and surveys are required for future developments to the application.

2.11 Requirements

The adapted requirements are based on Mann et.al (2019) and Ramachandran (2018)'s work:

- It should be a web-based application that is able to run on a PC,
- The application should be capable of displaying Tamil othography to represent the spoken Tamil word.
- The application should be able to recognise, and understand sound waves produced by the speaker in order to pick up the original word.
- The application should be able to interpret and understand real words and ignore words with errors or mispronunciation.
- Since the application uses a microphone, it must understand what to ignore and what to accept

2.11.1 Does the application meet the requirements?

Below are the functionalities of the application which are in line with the provided requirement specification document of the client.
a) The application consists a configured list of 28 Tamil words only and any other word would be ignored.
b) If the word is mispronounced then it would be ignored too
c) If a word from a different language is spoken, the application will ignore it and as a result no text will be displayed
Hence, the user will not face issues with converted text due to code-mixing and code switching as the application is designed to ignore those instances, and just convert recognized Tamil words to text.

2.11.2 Non-functional Requirements

According to Scaled Agile [13], non-functional requirements or 'NFR'define system attributes such as security, reliability,

performance, maintainability, scalability, and usability.'' Non-functional standards such as responsiveness, security, availability, reliability, usability, and many more are used to judge success of a software system. Example for non-functional requirement could be, "does system prevent any unauthorized access?"
It is not mandatory to include non-functional requirements in a software system, but its practice ensures good user experience and ease in operating the software. Functional requirements are used to describe what the product does whereas Non-functional requirements describes how the product works.
Drawbacks of Non-functional requirements are increase in cost as they require high-level design phase, they are difficult to modify once the architecture phase is completed [14].

3 METHODS

3.1 Software Engineering Methodologies

When considering the project, the right project management methodology was selected in order to ensure that the project is successful and completed on time.
A project management methodology can be defined as a set of techniques/methods/procedures or best practices used to complete a project, which are normally based on specific management approaches [15]. There are 2 different types of software development methodologies which are:
Predictive/structured methodologies – examples are such as the waterfall model, spiral model and the RUP (Rational Unified Process) model

Adaptive methodologies – Agile methodology
When selecting a software development methodology, it is important to remember that there is no best or one-size-fits-all methodology. Rather, the needs of the client or organization have to be assessed, and the best decision has to be taken based on the various factors of the project such as user requirements, team size, risk, cost, time and scope.

Waterfall methodology – The waterfall method iterates that software development follows a series of steps, like the cascading water of a waterfall [16]. The characteristics of the waterfall methodology are:

a. Every stage of the project from analysis to support is only done once
b. The project transitions from one stage to the next sequentially when the predetermined /objectives are achieved.
c. At the end of each stage, the deliverable is not the software, rather the documentation that reflects the goals/milestones of the work undertaken (e.g. business requirements or design [17].

Agile Advantages	Waterfall Disadvantages
Suitable for larger organizations as forces structured processes	Lack of adaptability, can't handle change
Allows for design changes early in the process	Poor visibility, as working software is only delivered at the end of the project
Clear requirements are gathered from the start	Risk of poor quality
	Ignores mid-process customer/client feedback
	Risky, as testing is conducted only near the end of the project

Agile method – The Agile methodology is based on an adaptive lifecycle or processes and is built to adapt to change [18]. Tasks are broken up into different stages or iterations between 1-4 weeks, with each iteration running a full development cycle (Planning, requirements, design, coding, testing). Working software is produced at the end of each iteration, with functionality delivered incrementally in stages. It is customer driven and emphasizes on customer collaboration throughout the project.

The Agile manifesto is as below:
"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

a. Individuals and interactions over processes and tools.
b. Working software over comprehensive documentation.
c. Customer collaboration over contract negotiation.
d. Responding to change over following a plan.

That is, while there is value in the items on the right, we value the items on the left more." [19].
The characteristics of an Agile project are:

a. Appropriate for uncertain or changing requirements
b. Stories and face to face communication daily as requirements of running an Agile project.
c. The aim is to deliver working software at the end of every iteration.

Advantages of the Agile methodology:

a. Improved quality, as testing is done from the beginning of the project.

b. Reduced risk, as customer feedback is obtained throughout the project.
c. High adaptability, changes can be easily made.
d. Improved visibility, as the project is halfway complete when 50% of the software is built.

For this particular project, the Agile methodology was initially selected to run the project based on several reasons:
Team size – Agile methodology is suited for smaller teams of between 7-9 members
Risk – Can be addressed throughout the duration of the project
Client interaction – An Agile project focuses a lot on customer interaction, which is something our group has been doing from the first day. The stakeholder is able to review progress and ensure the group is aligned to the project goals
Adaptability – As the project is customer driven, changes can be made if customer requests it.

However, due to several challenges while developing the app, the group resorted to a mixture of methods from both the waterfall and Agile methodology to complete this project.

Only one team member had experience in coding, hence it was not possible for a working version of the application to be produced at each iteration therefore the application was only able to complete at the end of the project duration.

Waterfall methods used – The app was developed in clear, defined stages. Working software was only produced at the end of the project, not in stages as per the Agile methodology. Testing was also done at the end of the project, upon the completion of the app.

Agile methods used – Regular client interaction & feedback, regular sprints (meetings), each team member had their roles and responsibilities clearly defined and project was completed with a small team comprising of 7 members.
During the early days of project management methodologies, only one methodology was to be used to complete a project, however for our project a mix of both methodologies or what's known as a hybrid methodology was found to be the key to helping us complete it sucessfully

Further to that, it is opined that it is possible and, in some cases, advisable to combine both traditional and adaptive methodologies for completing a project, keeping in mind at which stage is more compatible with which methodology [15].

Factor	Predictive	Adaptive (Agile)
Size	Methods evolved to handle large products and teams.	Well-matched to small products and teams.
Criticality	Methods evolved to handle highly critical products. Hard to tailor down to low-criticality products.	Untested on safety-critical products. Potential difficulties with simple design and lack of documentation.
Dynamism	Big Design Up Front excellent for highly stable environment.	Simple design and continuous refactoring are excellent for highly dynamic environments.
Personnel	Needs scarce expertise during project definition, but can work with fewer later in the project.	Requires continuous presence of a critical mass of scarce expertise.
Culture	Thrives in a culture where people feel comfortable and empowered by having their roles defined.	Thrives in a culture where people feel comfortable and empowered by clear processes.

Figure.4: Balancing Agility and Discipline: A Guide for the Perplexed (Boehm, B. 2009)

4 DEVELOPMENT

4.1 Technology Choice

This group developed a web-based application that is accessible on all Internet of Things devices, or IOT's. There are several programming languages to support this web-based software, but the group has chosen Java as it provides features like cross platform support, better performance in terms of speed when compared with python and is platform independent.

The following technologies were used for developing the application:

- Java
- MySQL
- Apache Server
- Spring MVC Model
- Hibernate
- JSP
- Servlet

The following scripting languages were used:

- Html
- CSS
- JavaScript
- JQuery
- Ajax

The feature for saving the synthesized wave file and for reading the existing text in TAB format is also provided.

4.2 Tamil speech to text conversion

Text conversion from spoken Tamil to written Tamil is perfomed through Google's cloud conversion API. This API is called from the server side, since the translated text needs to be further processed through by the Tamil speech to text engine.

Difference with existing systems

There are existing applications that perform speech to text conversions in Tamil but from the perspective of education particularly in pronunciation training they are inadequate as even with the wrong pronunciation provided the system tends to pick up the right words and autocorrects them. From a native Tamil speaker's perspective, this does not fulfil the requirement of *what you speak is what you get*, and instead helps to encourage mispronunciations and errors. As our application is aimed at Tamil speakers, the home page language was changed from English to Tamil and vice-versa for ease of usage and to conform to a design that is consistent with indigenous culture.

5 Testing and Verification

The prototype was due to be tested at each iteration of the development, with live testing carried out by the two native Tamil speakers who are part of the team. Due to the closure of the university and physical teaching facilities caused by the Covid-19 pandemic, the team revised the testing strategy as face-to-face testing was no longer possible.

The hardware used such as the microphone and speaker were tested before the app testing commenced. External noise and any factors that may affect the quality of the recording speech was taken into consideration. At the final stage of the project, a native Tamil speaking team member tested the system by speaking pre-selected words in Tamil into the microphone to verify the resulting text.

Other applications may have similar features but the use case for this application differs in the sense that it is only supporting a few words in the literature & not recognizing other words, and also not supporting auto-correction so the expectation is to speak the word using its correct pronunciation. Therefore, for these specific cases, the group decided to create a new application.

5.1 Results and analysis

To evaluate the general performance, perceptive assessment was carried out. The testing phase involved making use of native Tamil speakers as users and also to rate the synthesized speech in terms of intelligibility, naturalness and distortion on the different sentences. Each user was required to click the record button and speak a Tamil word. The utterance was recorded, and the audio file was sent to our server, which perfomed the speech recognition task and passed the recognized Tamil Unicode text to Google convert API. The recognized Tamil text and its synthesized waveforms were then transferred to the client-side and made available for viewing.

The system was tested against a configured list of 28 Tamil words which were used in the development of the prototype and application.

Tamil orthography	**Roman orthography**
அலை	Alai
முகநூல்	Muganool
அலை (அழை)	Alai (Azhai)
அளைப்பிதல் (அழைப்பிதழ்)	ALaippithal (azhaippithazh)
விளுப்புரம் (விழுப்புரம்)	ViLuppuram (Vizhuppuram)
கல்லு	Kallu
கள்ளு	KaLLu
பிளை (பிழை)	PiLai (pizhai)
வாலு	Valu
வால்க்கை (வாழ்க்கை)	vaalkkai (vaazhkkai)
வன்னம் (வண்ணம்)	vannam (vaNNam)
விலக்கு (விளக்கு)	vilakku (viLakku)
பலனி (பழனி)	Palani (Pazhani)
மாட்டு	Maattu
மீட்டு	Meettu
காலை	Kaalai
தொழிலாளி	ThozhilaaLi
கலை (களை)	Kalai (kaLai)
முட்டு	Muttu

கிளை	KiLai
குட்டு	Kuttu
கிளி (கிழி)	KiLi (Kizhi)
மேட்டு	Maettu
தொகுப்பாலர் (தொகுப்பாளர்)	Thoguppaalar (ThoguppaaLar)
எண்ணை	ENNai
ஏற்றுக்கொள்	EttrukkoL
எளுதுகோள் (எழுதுகோல்)	ELuthugoL (Ezhuthugol)
மனப்பான்மை	Manappanmai

The words in the brackets indicate the correct spelling and the correct pronunciation. The words outside the brackets indicate the wrong spelling but is consistent with the mispronounced word. For example:
The correct spelling that is consistent with accurate pronunciation of the word *Pazhani* in Tamil is பழனி. However, **ழ** in some cases are mispronounced by the native speakers as **ல.** In such cases, the output must be displayed as பலனி instead of பழனி. It is observed that the developed prototype does not display the former in this case. We assume that this is because of Google API.

The system was deliberately designed not to display anything given the following scenerios:

1) When the user mispronunced the word.
2) When the user spoke words other then those in the speech corpus provided by the client
3) When user spoke a different language

The system was tested by collecting pre-recorded audio from 2 Tamil speaking team members and those recordings were used for testing the application. A non-Tamil speaking member of the team also provided a pre-recorded audio file for testing purposes; however, the desired output was not displayed as the application wasn't able to distinguish the spoken words due to the user's accent. This provides an empirical evidence that in designing and testing a language based applications, the knowledge of target language including the nuanced pronunciation and the ability to distinguish between the correct and incorrect pronunciation become increasingly important and must be accorded a higher priority.

As illustrated below, the given yes or no outputs indicated whether the system displayed the correct word.

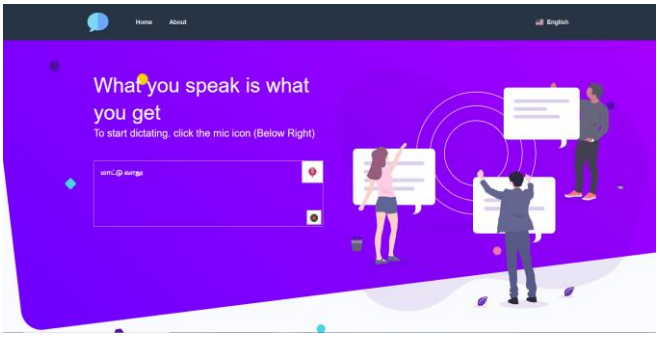
I21 fx

	A	B	C	D	E
1	Tamil orthography	Roman orthography	Ayush	Dheenesh	Logesh
2	அலை	Alai	Yes	YES	YES
3	முகநூல்	Muganool	Yes	YES	YES
4	அலை (அழை)	Alai (Azhai)	Yes	YES	NO
5	அளைப்பிதல் (அ	ALaippithal	Yes	YES	NO
6	விளுப்புரம் (விழு	ViLuppuram	Yes	NO	YES
7	கல்லு	Kallu	Yes	YES	NO
8	கள்ளு	KaLLu	Yes	NO	NO
9	பிளை (பிழை)	PiLai (pizhai)	Yes	YES	NO
10	வாலு	Valu	Yes	NO	NO
11	வால்க்கை (வாழ்	vaalkkai (vaazhkkai)	Yes	YES	YES
12	வன்னம் (வண்	vannam (vaNNam)	Yes	YES	NO
13	விலக்கு (விளக்கு)	vilakku (viLakku)	Yes	YES	YES
14	பலனி (பழனி)	Palani (Pazhani)	Yes	YES	YES
15	மாட்டு	Maattu	No	YES	NO
16	மீட்டு	Meettu	No	NO	YES

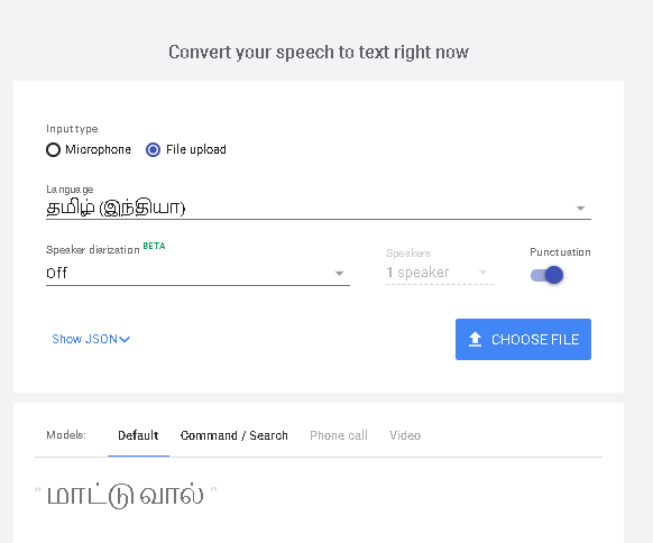
Figure 5: Test case

To display the difference between this application and a similar speech to text application, a short test scenario was devised using 2 Tamil words chosen at random from the 28 provided in the corpus, to see if any differences would occur in the conversion. For comparison purposes Google's Speech to Text was used. The results are displayed as below:

Example of the prototype:



Output from the Google's Speech to Text:



As is evident from the two screenshots, the displayed output differs slightly in the characters displayed. In this case any variation in the text as is displayed in Google's speech to text is incorrect. We argue that this is also largely dependent upon the consistency in pronunciation as even a slight variation in pronunciation could result in a different spelling thereby altering the result.

6 Discussion

This project was aimed at developing a web-based speech to text application that enabled users to convert spoken Tamil into text. The application achieved one of its core requirements of converting spoken speech to text using an application that can also be used to act as a teaching tool to confirm correct pronunciation of syllables and words for native and non-native Tamil speakers. In essence, the user was able to speak a number of Tamil words from the pool of 28 words into the application and receive accurate text in Tamil orthography. However, the application could not produce any output when incorrect pronounciation of the word in question occurred. This continues to be a major area of research and focus.

The users were able to speak into the web-based application and managed to obtain text in Tamil. The COVID 19 pandemic provided us a unique opportunity to explore alternate methds of testing, as testing the application in a live setting with the testers was no longer possible. We overcame this by the use of pre-recorded audio files from our designated testers within the group and it was successful as the expected output was displayed by the application on-screen.

7 Conclusion and Future Directions

The research has attempted to develop a speech to text application using Google's cloud conversion API, which can be used to convert spoken Tamil speech to Tamil text, according to client requirements. An ethnically diverse team has helped approach the project from several viewpoints and has provided a unique set of skills to complete this project. This web-based application is useful for users who wish to learn and practice the proper pronounciation of Tamil words. The team has built ASR and used Google's speech to text translation tool to build this application. The project will enable the client to spread and increase awareness and usage of the Tamil language by using it for educational purposes. Attempts are being made to make it natural, add emotions and making it net enabled. The system is extendable to any other language just by changing the language rules, intonation and the database. This research through this work also emphasizes on the indigenious design considerations for such applications.

Further work on this software may include developing a corpus rather than using API and developing it on a mobile platform, as a larger number of people will be able to access and use it. Developing a mobile version of this application may encourage the use of the app in various other scenarios such as when texting and in an educational setting.

Further work on the application would include the ability to recognize and convert any word in Tamil, instead of the 28 words that are specified in the requirements as a part of this research. The team also aims for the application to provide an error message when any code-mixing or code-switching is detected, and if any words from a different language are spoken into it.

References

[1] Fearn, N, Turner, B. (2020). *Best speech to text software in 2020: Free, paid and online voice recognition apps and services.* Available from: https://www.techradar.com/uk/news/best-speech-to-text-app [Accessed 18 April 2020]

[2] Tebelskis, J. (1995). "Speech Recognition using Neural Networks". CMU -CS -95 -142, School of Computer Science, Carnegie Mellon University Pittsburgh, Pennsylvania 15213-3890.

[3] Sakai, K.L., 2005. Language acquisition and brain development. *Science*, *310*(5749), pp.815-819.

[4] Ferguson, C.A., 1959. Diglossia. *word*, *15*(2), pp.325-340.

[5] Keane, E. (2004). Tamil. Journal of the International Phonetic Association, 34(1), 111-116.

[6] McGuire, N. (2018). *How Accurate is Google Tranlate?* Available from: https://www.argotrans.com/blog/accurate-google-translate-2018/ [Accessed 18 March 2020]

[7] Orchilles, J. (2010). *Speech Recognition.* Available from https://www.sciencedirect.com/topics/computer-science/speech-recognition. [Accessed 29 January 2020]

[8] Das, Prerana & Acharjee, Kakali & Das, Pranab & Prasad, Vijay. (2015). VOICE RECOGNITION SYSTEM: SPEECH-TO-TEXT. Journal of Applied and Fundamental Sciences. 1. 2395-5562.

[9] Stephanie (2019) *The Current Challenges of Speech Recognition.* Available from: https://onlim.com/en/the-current-challenges-of-speech-recognition/ [Accessed 29 January 2020].

[10] Madhavaraj and A. G. Ramakrishman, "Design and development of a large vocabulary, continuous speech recognition system for Tamil," *Proc. 14th IEEE India Council International Conference (INDCON)*, Dec 2017.

[11] Fast Typing (2020). Tamil Typing Speed Test. Available from https://www.fast-typing.com/typing-speed-test.php?lan=tamil [Accessed 29 January 2020].

[12] Way, A., 2018. Quality expectations of machine translation. In Translation quality assessment (pp. 159-178). Springer, Cham.

[13] Scaled Agile. (2018). Nonfunctional Requirements. Available from https://www.scaledagileframework.com/non-functional-requirements/

[14] Glinz, M., 2007, October. On non-functional requirements. In *15th IEEE International Requirements Engineering Conference (RE 2007)* (pp. 21-26). IEEE.

[15] Špundak, M., 2014. Mixed agile/traditional project management methodology–reality or illusion?. *Procedia-Social and Behavioral Sciences*, *119*(1), pp.939-948

[16] Powell-Morse, A. (2016). *Waterfall Model: What is it and when should you use it?* Available from https://airbrake.io/blog/sdlc/waterfall-model. [Accessed 01 February 2020]

[17] Ahimbisibwe, A., Cavana, R.Y. and Daellenbach, U., 2015. A contingency fit model of critical success factors for software development projects. *Journal of Enterprise Information Management*.

[18] Sheffield and Lemétayer, 2013

[19] (Agile Manifesto 2001)

[20] Leonard, K. (2019). *Good Typing Speed per Minute.* Available from: https://smallbusiness.chron.com/good-typing-speed-per-minute-71789.html [Accessed 01 February 2020]

[21] Manish, S. (2019). *How Hindi's growth will lead to decline in Tamil, Kannada speakers.* Available from: https://www.business-standard.com/article/current-affairs/how-hindi-s-growth-will-lead-to-decline-in-tamil-kannada-speakers-119092400536_1.html [Accessed 20 April 2020]

[22] Muthusamy, P. and Farashaiyan, A., *A Descriptive Study of Tamil Language and Social Identity Among English-Tamils in Malaysia*. Available from: https://www.researchgate.net/publication/326985503_A_Descriptive_Study_of_Tamil_Language_and_Social_Identity_Among_English-Tamils_in_Malaysia [Accessed 30 April 2020]

[23] O' Dea, S. (2020). *UK households: ownership of mobile telephones 1996-2018* Available from: https://www.statista.com/statistics/289167/mobile-phone-penetration-in-the-uk/ [Accessed 31 March 2020]

[24] Sigappi, A.N., Palanivel, S. and Ramalingam, V., 2011. Handwritten document retrieval system for Tamil language. *Int. J of Computer Application*, *31*.

[25] Singh, B., Kapur, N. and Kaur, P., 2012. Speech recognition with hidden Markov model: a review. *International Journal of Advanced Research in Computer Science and Software Engineering*, *2*(3).

[26] SRIGANESH, B., *Beyond Munneru Vaaliba: Reinvigorating the Tamil language in Singapore*. Available from: https://spj.hkspublications.org/2018/02/25/beyond-munneru-vaaliba-reinvigorating-the-tamil-language-in-singapore/ [Accessed 30 April 2020].

[27] Stein, B., 1977. Circulation and the historical geography of Tamil country. *The journal of Asian studies*, *37*(1), pp.7-

[28] Thangarajan, R., Natarajan, A.M. and Selvam, M., 2008. Word and triphone based approaches in continuous speech recognition for Tamil language. *WSEAS transactions on signal processing*, *4*(3), pp.76-86.

[29] The Star, (2007). *Usage of Tamil language fast declining*. Available from: https://www.thestar.com.my/news/nation/2007/05/07/usage-of-tamil-language-fast-declining [Accessed 26 February 2020]

[30] Mann, D., Weston, N., Frederic, K., Ogunshile, E., & Ramachandran, R. (2019, October). Tamil talk: What you speak is what you get!. In *2019 7th International Conference in Software Engineering Research and Innovation (CONISOFT)* (pp. 36-48). IEEE.

[31] Ramachandran, R., 2018. *Predicting user acceptance of Tamil speech to text by native Tamil Brahmans* (Doctoral dissertation, Sheffield Hallam University (United Kingdom)).