DOI: 10.15514/ISPRAS–2021–33(1)–13

A reflection on the design and user acceptance of *Tamil talk*

R. Ramachandran, ORCID: 0000-0003-0355-698X <raj.ramachandran@uwe.ac.uk>
E. Ogunshile, ORCID: 0000-0002-6276-188X <Emmanuel.ogunshile@uwe.ac.uk>
University of West of England,
Coldharbour Ln, Bristol, BS16 1QY UK

Abstract. *Tamil talk* is a speech to text application and was designed from a perspective of language and philosophy. This paper takes an indigenous approach in reflecting on the design and user acceptance of *Tamil talk*. The paper makes use of literature in critically reflecting on the design and the potential user acceptance of the application. It takes a multi-disciplinary approach and explores the influence of factors like language shift, language maintenance and philosophy in the context of user acceptance of speech to text. The application may appeal to a section of the native Tamil speakers as suggested in the literature but there are complex challenges that needs further research. Further research shall be in developing the application that conforms to the conceptual framework and widely test with the native speakers to arrive at a more precise prediction of user acceptance.

Keywords: Speech to text; Use acceptance; Philosophy; Tamil; Software Engineering

For citation: Ramachandran R., Ogunshile E. A reflection on the design and user acceptance of *Tamil talk*. Trudy ISP RAN/Proc. ISP RAS, vol. 33, issue 1, 2021, pp. 189-208. DOI: 10.15514/ISPRAS–2021–33(1)–13.

Размышление о дизайне и восприятии пользователями приложения *Tamil talk*

Р. Рамачандран, ORCID: 0000-0003-0355-698X <raj.ramachandran@uwe.ac.uk>
Э. Огуншиле, ORCID: 0000-0002-6276-188X <Emmanuel.ogunshile@uwe.ac.uk>
Университет Западной Англии,
Великобритания, BS16 1QY, Бристоль, Колдхарбор Лейн

Абстрактный. *Tamil talk* – это приложение для преобразования устной речи в текст, разработанное с позиций языка и философии. В этой статье используется автохтонный подход к осмыслению дизайна и принятия пользователями приложения *Tamil talk* на основе анализа литературных источников. Используется междисциплинарный подход и исследуется влияние таких факторов, как смена языка, приверженность языку и философия в контексте принятия пользователем преобразования устной речи в текст. Как полагают авторы литературных источников, такое приложение может импонировать части носителей тамильского языка, но имеются сложные проблемы, которые требуют дальнейшего исследования. Дальнейшие исследования должны быть направлены на разработку приложения, соответствующего концептуальной модели и испытываемого большим числом носителей языка, чтобы прийти к более точному пониманию принятия пользователями этого приложения.

Ключевые слова: преобразование речи в текст; принятие пользователями; философия; тамильский язык; программная инженерия.

Для цитирования: Рамачандран Р., Огуншиле Э. Размышление о дизайне и восприятии пользователями приложения *Tamil talk*. Труды ИСП РАН, том 33, вып. 1, 2021 г., стр. 189-208 (на английском языке). DOI: 10.15514/ISPRAS–2021–33(1)–13

1. Introduction

Speech is the ability to express thoughts and feelings by articulating sounds. It is a key component of communication for humans. Each language uses phonetic compositions of a limited set of vowels and consonant sounds that form words. Speech recognition is the interdisciplinary subfield of computational linguistics that develops methodologies and technologies that enables the recognition and translation of spoken language into text [1]. There are over 6000 languages spoken in the world. Everett suggests that humans can learn all the sounds they want when they are young. The articulation of sounds gives rise to speech forms. Some languages share similarities whilst some don’t. Accuracy of pronunciation by the speaker could be argued as a requirement in languages where there is a one to one correspondence between the sound and orthography. In cultures that consider philosophy as a way of life, it is even more important to consider the philosophical views and practices on how a language ought to be spoken amongst other factors. The research explores the feasibility to develop a speech to text application using a native approach and a conceptual framework of ‘what you speak is what you get’ with a focus on user acceptance.

This paper is an extension of [2] and builds upon the work of [3]. This paper reflects on Tamil talk – an application that was developed to convert spoken Tamil speech into Tamil orthography with a conception view of ‘What you speak is what you get’. It attempts to discuss the user acceptance of Tamil talk using secondary sources such as ethnography, in particular, the paper explores the link between the use of language at social sphere and the behaviour intention to use that language in the technology such as speech to text. It sets focus on areas like language shift, language proficiency and the challenges associated with it in using Tamil talk.

Tamil is a syllabic language with almost one to one correspondence to the sound and orthography [4]. The work of [5], point to unique syllables of Tamil often mispronounced especially the syllable 'zha' by the native speakers for various reasons as also identified and raised in [3] in the context of not just predicting the user acceptance of speech to text but also to be able to use speech to text in Tamil owing to the nature of the language. There are very few studies on recognition of 'zha' as seen in [6], and [7]. It is argued that the views as in [8] on language variation do not apply in this context due to the syllabic nature of the language.

The first successful speech recognition machine was released by three Bell Labs researchers in 1952 and it was called Audrey. This machine was able to recognize spoken digits with 90% accuracy; however, it only recognized the inventor’s voice [9]. The technology then developed into understanding English words about 10 years later. Raj Redy was one the first to research on continuous speech recognition as a graduate student in the late 1960s. Systems prior to this, required users to pause after every word.

Speech recognition technology has grown in sophistication and accessibility leading to the ability to convert speech into text for hundreds of languages and dialects. This includes languages like Tamil, which uses a script that is different to Roman script. Nowadays, almost everyone has a smartphone which is more than capable of recognizing spoken language using the most basic of hardware. The research conducted in this paper points to different design possibilities and details of developing a speech to text application, more specifically in a non Roman script. An application was built to convert spoken Tamil to Tamil text and display what has been spoken onto the screen of a given device. Example: A smartphone or a laptop. The paper makes use of literature as a framework to reflect on the design and potential acceptance of the application on the basis of the literature primarily from the perspective of philosophy and cultural anthropology.

The aim of this research is to predict the user acceptance of Tamil talk developed on the conceptual framework as in [2] ‘what you speak is what you get’

The application itself was designed to work as a teaching tool for native and non-native Tamil speakers to confirm correct pronunciation of syllables and words.

The structure of the paper is as follows:

- The first section deals with the research carried out into the Tamil language and investigates any

speech to text applications that are currently available.
- The second section describes the method used to design, develop, implement and test the application developed.
- The third section documents the testing carried out on the developed application.
- The final section provides the discussion and conclusion for the project and suggested work that could be conducted in the future.

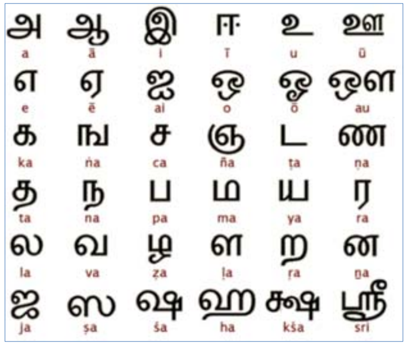
2. Literature Survey

Speech to text is the conversion of spoken words into text. ASR applications are very widely available and help assist millions of people every day, however, many of these applications have language-based limitations by only supporting a handful of languages. Developing an application that supports "minority languages" would provide great technological benefit and help develop more inclusive and accessible technology [10]. Minority language is defined as a language spoken by less than 50% of the given region, state, or community [11]. This research views the definition of minority language as seen in [12] outside of the context and region where Tamils are in majority, for example, in the Indian state of Tamil Nadu and Puducherry.

2.1 Introduction to Tamil

Tamil is a Dravidian language spoken by eighty million people in South Asia and across fifty-five diaspora countries but predominantly by the Tamil people of Tamilnadu [12]. It is one of the longest surviving classical languages in the world and recorded Tamil literature has been documented for over 2000 years [13].

The Tamil script consists of 12 vowels (Uyir Ezhuthukkal), 18 consonants (Mei Ezhuthukkal) and one special character, the Aytha Ezhuthu (see fig.1). The vowels and consonants combine to form 216 compound characters (Uyir Mei Ezhuthukkal), giving a total of 247 characters [14]. Unlike other South Asian scripts, such as Gujarati, Tamil does not have script to represent voiceless aspirated (such as "kh"), voiced ("g") and voiced aspirated stops ("gh").



அ	ஆ	இ	ஈ	உ	ஊ
a	ā	i	ī	u	ū
எ	ஏ	ஐ	ஒ	ஓ	ஔ
e	ē	ai	o	ō	au
க	ங	ச	ஞ	ட	ண
ka	ṅa	ca	ña	ṭa	ṇa
த	ந	ப	ம	ய	ர
ta	na	pa	ma	ya	ra
ல	வ	ழ	ள	ற	ன
la	va	ẓa	ḷa	ṛa	ṉa
ஜ	ஸ	ஷ	ஹ	க்ஷ	ஸ்ரீ
ja	ṣa	śa	ha	kša	sri

Fig 1. Basic Tamil script [14]

2.2 Speech to Text in Tamil

There are many existing speech-to-text software and applications that support the Tamil language. The most commonly used system is Google Translate [15]; which could potentially be used to translate from one language to another, such as French to English, but could also arguably used for speech to text. Unfortunately, there are shortcomings with the existing application. The existing system does not incorporate the concept of 'what you speak is what you get'. The concept as seen in [3] is based on the structure of the language as opposed to an application that accommodates language and pronunciation variation of the users.

This kind of predictive speech-to-text will give the users a sense of pronouncing the words correctly, where in reality the words are mis-pronounced. This makes users believe they speak correctly and will teach them the language incorrectly which is a significant issue [16].

2.3 Design considerations for Tamil talk

In a standard speech recognition system (fig. 2) the raw speech is typically sampled at a high frequency (16KHz – 8KHz) this produces a sequence of amplitude values over time. This raw speech data is then transformed and compressed to enable simplified processing [17]. Many signal analysis techniques are available which can extract useful features and compress the data by a factor of ten, such as Fourier analysis, Perceptual Linear Prediction, and Lineal Predictive Coding.

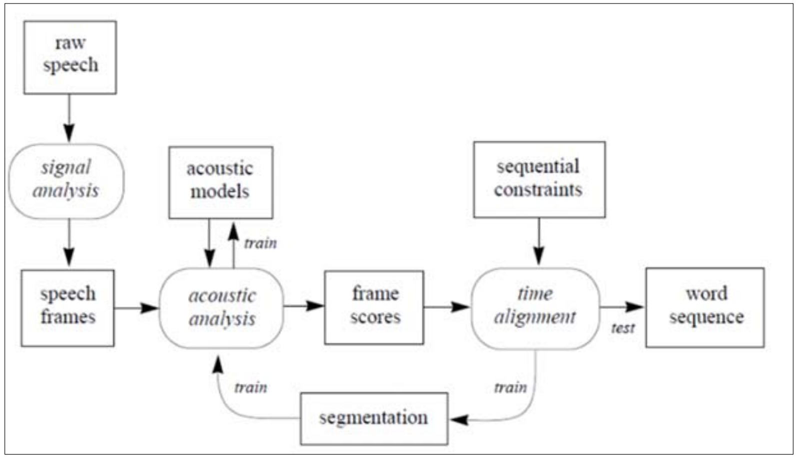


Fig. 2. Structure of standard Speech Recognition System [17]

The speech-to-text transformation is one of the most difficult tasks in computer science because it consists of many difficult problems as seen in [18] but it is an important technology to develop well. Speech recognition has many potential applications including command and control, dictation, transcription of recorded speech, searching audio documents, and interactive spoken dialogues [19]. At the core of all speech recognition systems consists of a set of statistical models representing the various sounds of the language to be recognized.

The task of speech recognition is to find the best matching word-sequence ($\hat{W}$) given the data of an utterance (O). O is a sequence of input vectors generated from the raw speech data. According to Bayes' Theorem the task can be formulated as seen below [20]:

$$\hat{W} = \arg\max_{W} P(W|O) = \arg\max_{W} \overbrace{P(O|W)}^{AM} \cdot \overbrace{P(W)}^{LM}.$$

To decode the sequence of words spoken you have to calculate the next probability where W is the decoded sequence and O is the observed sequence or incoming feature vector [20]:

$$\hat{W} = \arg\max_{W} P(W|O).$$

This splits the task into two components $P(O|W)$, also known as the acoustic model and $P(W)$,which is also known as the language model. In most speech recognition systems, the acoustic model is represented by the HMM or the Hidden Markov Model [21]. Each HMM is a finite state machine with n states whereby each state, besides the first and last, has specific output probabilities and each arc between states is associated with a transition probability [19]. Previously, the output probabilities were modelled by multivariate Gaussian Mixture Model or GMM [22]. Given the restraints in computational power, the GMM is restricted to have a diagonal co-variance matrix and hence require independence between the input dimensions.

Artificial neural networks (ANNs) on the other hand, do not need this requirement of independence, which is why they are more currently used for acoustic modelling and give increased performance to speech recognition, particularly Convolutional Neural Networks and Recurrent Neural Networks [23]. There are several possible ways to exploit ANNs in automatic speech recognition systems, such as the Hidden Markov Model-Artificial Neural Network hybrid system, which takes the advantage of ANN's strong representation learning power and HMM's sequential modelling ability.

There are researches around Connectionist Temporal Classification (fig. 3) which is very similar to the standard HMM-ANN approach however a single-state HMM is used, usually with three times lower output frame-rate and after each single-state HMM a "blank" state is allowed [24]. Further, this model does not need alignment of the training data, which is an advantage, but only starts to show benefits starting with very large quantities of data [25].

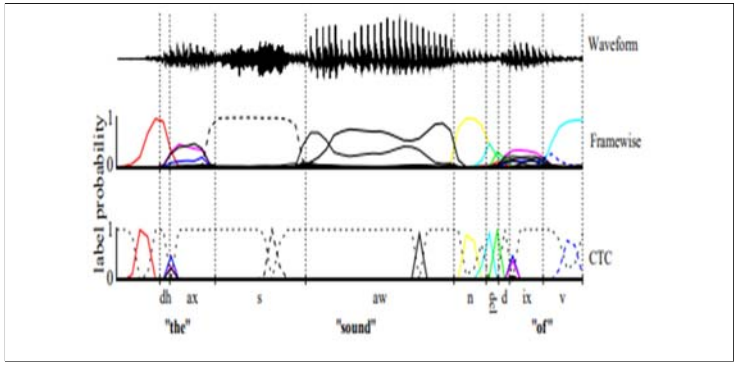


Fig 3. Frame wise and CTC networks classifying a speech signal [24]

All these options were considered at the beginning but we eventually worked on two options that was thought would provide a complete solution of creating a Speech-to-Text application in Tamil that uses the conceptual framework of "what you speak is what you get". While it was deemed the option to develop an application using the Dictionary Model would be the best fit for this project, both options have been described in sections 2.5 and 2.6, allowing for the information to be used for any enhancement to the solution at later date.

2.4 Application Programming Interfaces

An Application Programming Interface (API) is a set of subroutine definitions, protocols, and tools for building application software. In general terms, it is a set of clearly defined methods of communication between various software components. Many API's are built with the intention to allow third party developers to build interesting applications and designed to expand the reach of an organization [26].

All speech recognition engines/API's initially work in the same way as the user's voice is passed through the microphone input to reach the recognition system. There are a number of options for Automatic Speech Recognition API's available that can be divided into closed source code and open source code (fig. 4).

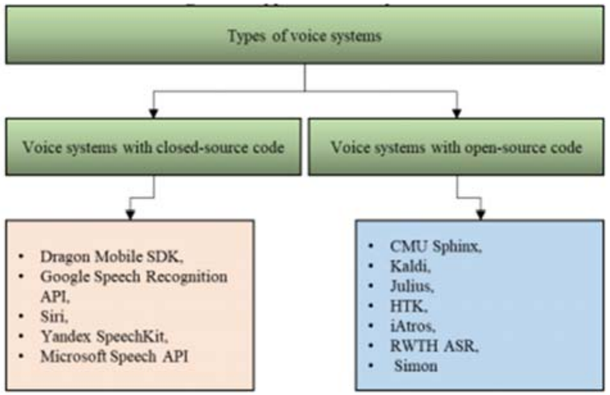


Fig 4. Types of voice systems [29]

Closed source code means that there is no physical access to the code and user will be unable to manipulate and modify it. Two of the biggest companies building voice-powered applications are Google and Microsoft; however, these API's code is inaccessible [27].

CMU Sphinx is one of the most renowned systems and is completely open source. Sphinx enables developers to download and use their code freely as in [28] which includes speech recognizers and acoustic model trainers [29].

There are many benefits of using API's when building applications, especially if the API meets the specific needs of an application, then avoiding the reinvention of the proverbial wheel is a standard piece of wisdom in software development circles [30]. If it is not a good fit for a proposed application but the API is open source and open code then users will be able to access and manipulate the code to fit with an application. Additionally, there are also disadvantages to API's that need to be considered. Anyone could have written the code and if it is closed source there is almost no way of knowing what it actually does and that limits the opportunity and ability to improve.

Google Speech Recognition API is a technology used widely in different research fields for different languages. It allows the user to voice search and its technology is integrated into many smartphones and computers. Originally it only supported a short request of a maximum 40 words and has only recently improved its speech recognition by using a new technology, deep learning neural networks [31].

Part of Googles improvement is a major new feature in the Speech-to-text API that now allows developers to select between different machine learning models [32]. The speech API now supports over 100 languages, including ancient languages such as Georgian (first spoken in 430AD), as well as Swahili, Gujarati and Tamil in a bid to make the internet more inclusive [33].

Google acquired several deep learning companies over the years such as, DeepMind, DNNresearch and JetPac and using these deep learning neural networks Google achieve an 8 percent error rate in 2015, a reduction of more than 23 percent from 2013 [34].

In 1986 Sphinx was launched and became one of the most successful open source systems developed for research purposes using HMM [35]. Developed at Carnegie Mellon University (CMU) it currently has one of the largest vocabularies and speaker independent recognition codebase. The speaker independent speech recogniser uses HMM and n-gram statistical language model, which is able to recognise continuous speech with a big vocabulary [29].

Since its initial release, Sphinx has gone through a number of different modifications. In the past, the decoding strategy of the Sphinx systems tended to be deeply entangled with the rest of the system. As a result of these constraints, the systems were difficult to modify for experiments in other areas [36].

However, the newest version, Sphinx-4, released in 2010, works with various kinds of language specifications such as grammars, statistical language models or blends of both [37]. This means a major benefit of the Sphinx system is that researchers are given more flexibility in the way they incorporate acoustic models allowing constraints to be imposed on the input from the user as seen in [38], making it a great way for identifying particular phonetic sounds for set phrases.

Sphinx has now developed into a toolkit that can be used to develop powerful speech recognition applications and includes several parts as in [39]:

1) PocketSphinx is small fast program, processing sound, acoustic models, grammars and dictionaries.
2) The library Sphinxbase is necessary for PocketSphinx work.
3) Sphinx4 is the recognition library.
4) Sphinxtrain which is for acoustic models training.

Microsoft's Speech API has been around since 1993, developed by three of the four people responsible for the CMU Sphinx-II speech recognition system [31]. Its system is very similar to Google Speech API but uses a server application programming interface (SAPI) for its data. It includes a set of effective methods and its data is well integrated into the .NET framework [40].

Microsoft has continued to develop the powerful speech API and has released a series of increasingly powerful speech platforms as seen in [41], focusing on increasing the emphasis on speech recognition systems and improved Speech API by using a context dependent deep neural network hidden Markov model [42].

432.5 User acceptance consideration

The model proposed by [3] to predict the user acceptance of speech to text application is based on the UTAUT (Unified Theory of Acceptance and Use of Technology) model. The work on language maintenance in Singapore provides a critical insight on the potential consequences arising from language shift. [43] also provides an insight on the relationship between identities and script. Arguably, the identities apart from Tamil speaking Muslims, the concept of script and identity could also be related to the Tamil speaking Brahmans [44] who speak a highly Sanskritised Tamil as pointed in [43], [44], [6]. The used of the term 'Sanskritised Tamil' suggests presence of Sanskrit vocabulary in Tamil arguably by a minority of the Tamil society. There are compelling reasons to consider the aspect of code mixing, code switching, accuracy of pronunciation, choices of orthography as emphasised in [3]. Code mixing, according to [45] are of three types: insertion or embedding, alternation between the structures, congruent lexicalisation.

On the other hand, Creole of the Indian ocean such as Mauritius presents interesting insight on the value of language, and the kind of transformation, the society undergoes over time. The usefulness of English in the economic front, motivated them to retain it as an official language unlike the Reunion which is typically French [46]. The work of [47] suggest the indigenisation of the West in Asian culture such as in Malaysia and Singapore. More importantly, it discusses, the negotiation of identity which we argue is crucial from the point of user acceptance of a language based technology as it is inextricably linked to the choice of language, orthography literacy and preference as seen in [3].

[48] provides example of Ramanujam who lived in the United States from 1959, wrote primarily in English despite deeply rooted in South Indian Brahman culture. The work of [44] provides an interesting insight to the choices made by Brahmans, Tamil Brahmans in their case. [49] presents an interesting studies around language, education from the global southern contexts. In their work, they explore the question "Can English function as the national tongue of the once-colonised nations?". The work of [50] brings out the struggle between the Tamil purists and 'others'. Mandarin, Tamil and Malay are important identity markers for the Chinese, Indians and Malays within Malaysia [51]. The work of [52] and [46] provide perspectives on Standard Spoken Tamil.

2.6 The Dictionary-Based Approach

The second available option was the Dictionary-based approach. This section investigates and discusses it based on the project. Everyday speech by human is referred to as spontaneous speech; it is not scripted and therefore adds a variety of phenomena to a speech recognition task with false starts, human and non-human noises, new words and alternative pronunciations. All of this has to be tackled when creating a system for spontaneous speech recognition [53]. Rather than looking for the "correct" pronunciation of a word the system should automatically expand and adapt the phonetic dictionary and choose a word according to its frequency.

A dictionary-based approach is a basic approach for a speech-to-text system [18]. The pronunciation and sounds are looked up in a sound wave dictionary, these dictionaries are usual modified by hand or by applying phonological rules to a given dictionary [53].

As seen in [53], the Dictionary Learning Algorithm was proposed for using both a speech and phoneme recognizer:

1) Collect all occurrences of each word/tuple in the database and run the phoneme recognizer on them using the smoothed phoneme LM
2) Compute statistics of the resulting phonetic transcriptions of all words/tuples
3) Sort the resulting pronunciation candidates using a confidence measure and define a threshold for rejecting statistically irrelevant variants
4) Reject variants that are homophones to already existing dictionary entries
5) Reject variants which only differ in confusable phonemes
6) Add the resulting variants to the dictionary
7) Test with the modified dictionary on the cross validation set (optional)
8) Retrain the speech recognizer, allowing the use of multiple pronunciations during training.
9) As an optional step corrective phoneme training can be performed
10) Test with the resulting recognizer and the modified dictionary on the cross validation set
11) Create a new smoothed language model for the phoneme recogniser, incorporating all new variants.
12) Optional second pass.

There are disadvantages to the Dictionary Approach especially when entries are added by hand as this usually focuses on single occurrences of a word and it can introduce a number of errors into the system [18]. When modifying words in a dictionary by hand it is important to be aware that experts tend to use the "correct" phonetic transcription of a word which is not necessarily the most frequent or even the most likely transcription for a given task. The actual pronunciations may also be very different for the pronunciation deemed as "correct", and in spontaneous speech there are a lot of alternative pronunciations and they are not easy to predict. The phonetic dictionary is one of the main knowledge-sources for speech recognition but it is still regarded as being less important at acoustic or language modelling. As we have seen in speech recognition research the emphasis is often on the correct pronunciation of a word as it can be found in a lexicon, but this correct pronunciation does not have to be the main feature and does not provide the best recognition accuracy.

The Dictionary-Based method proposes a solution to reduce the system complexity, but this method will fail if a word looked up cannot be found in the systems dictionary. However, for this application this method should provide the user with the accuracy of their speech and not the "correct" spelling of the word as specified in the requirements.

3 *Methods*

Selecting the most appropriate development methodology is not easy. For this project there were two considerations that were priority, the type and complexity of application that was being developed, stakeholders, timescale and the experience of the development team.

3.1 Requirements and design

This subsection identifies the requirements expected to be met to produce a successful product for the end client. The requirements must be clear, complete and understandable to the stakeholders so that all parties are aware of what the system should be doing and any constraints to the software development process.

The method of project management chosen by the team was agile, so the approach to the development was iterative. The application itself was quite simple in design, meaning there are very functions:

a) It should be a web application that is able to run on a PC, smartphone or a tablet.
b) The application must be able to print Tamil orthography to represent the spoken Tamil word.
c) The application must be able to interpret and understand sound waves produced by the speaker in order to pick up the original word or sound.
d) The application must be able to interpret and understand real words and ignore words with errors or mispronunciation.
e) As the application uses a microphone, it must understand what to ignore and what to accept, meaning if there is 2 or more people speaking or there is background noise, it must focus on just the user.

The above are the functional requirements that entail the capabilities, appearance and interactions this system has with the users also known as the target audience for this project. They were measured during the testing and verification stage which is detailed further on in the paper.

The following are the non-functional requirements for the application:

a) The application must be able to recognize the spoken Tamil word to the written Tamil word in real time.
b) The application must be available for use by several users simultaneously.
c) As the application is web based, users should be able to access the application on any internet browser on any device, whether it be Linux, Windows, iOS or Android.
d) The application should achieve a high accuracy rate for translation from speech to text.
e) The application will provide the user with clear options of how to use the features.
f) The application shall have a simple design that only displays the necessary features.

The design process started with each member researching applications already available. This research was discussed during initial meetings with the full group and the sponsor. A high-level idea of the layout was put together. From this a selection of mockups were created using photoshop, and other design software, and an initial back-end prototype was developed calling on Google's speech recognition software. Further web design mockups were created and then compiled to create a high-fidelity prototype which was demonstrated to the sponsor where feedback was obtained and designs further modified.

This helped us steer in the right direction as to how visually the application should look and this was consistently implemented throughout the design and development stages. This technique proved to be the beneficial as it relied on the perspective of many people.

To visualize the application a series of use cases/user stories were created, to map out the user journey and information flow through the application.

3.2 Implementation

The tools used for initial implementation on this application were CMU Sphinx Python-based libraries, SQL Database and Javascript for the front end. These tools were chosen be the structure as they are able to interlink with one another to create a functioning application.

CMU Sphinx and Pocket Sphinx are the best source for this application as it was designed to support such a project.

3.3 Architecture of *tamil talk*

The following section deals with the application from both the back end and client front end.

3.3.1 System Architecture

Originally the system needed to be in a format that was shareable with others and to provide this, it was decided the software will be created in a mobile application format sharable on the Android Play Store or iOS app store. Subsequently, the requirements changed and the application was designed to be web-based. The following technologies were decided to be used:

- Python;
- Ionic Hybrid web application framework;
- CMU Sphinx PocketSphinx API;
- Apache Server;
- MySQL;
- MVC Design Pattern.

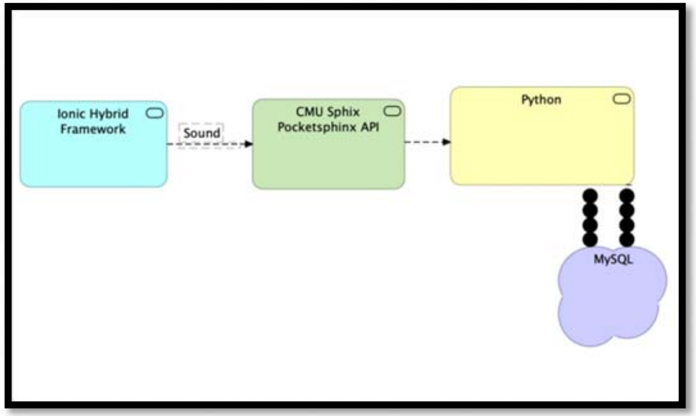


Fig 5. TamilTalk Architecture

To summarize fig. 8, the initial design of the web-based application had a MVC design with an Ionic based front end that communicates with a back-end written in Python, while using the PocketSphinx API to convert the sound into text.

3.3.2 HTML, CSS & Javascript

Due to unforeseen circumstances the original design model was not able to be fully developed. To ensure that a working software was developed for testing another approach had to be adapted. Time constraints placed an urgency on the development process, with this in mind a simple API implementation was used for the application. The Google Web Speech API was the choice of developers as it allowed for an efficient build of a working web application. This API provides a prebuilt dictionary of Tamil syllables, words, and orthography unique to the four major regions where the language is spoken (India, Sri Lanka, Malaysia, Singapore).

The main functionality was written in JavaScript, so it allowed easy implantation into a web browser. The graphical user interface was transferred over from the original design as it was built using simple HTML and CSS. Several constraints would arise while using the Web Speech API to provide the back-end functionality. The Web Speech API's sever-side language is not open source, so ensuring the correct process of conversion from speech to text became a difficult task. Additionally, this API would require all potential users to have an internet connection as well as use the latest version of Google Chrome, as it is the only web browser that works with this API.

4. Testing and verification

Testing was carried out to help define whether the requirements were achieved and this was implemented in the final stages of the development iteration.

Design protocol testing is imperative in software development as it will help entail how achievable the requirements are. The non-functional requirements were required to be successful and to demonstrate the quality level of the project.

This verification method was conducted by individuals of the team that were not associated with the development of the 'coding aspect' of the software to achieve an unbiased result. Other forms of verification were to monitor the risks associated with the development of the software on a continual basis to measure the success of the application; more risks entailed a higher failure rate. Feedback on initial design, storyboards were provided. The feedback was then incorporated into the design of the application showing the potential that it could be taken much further in the future.

In the testing phase the application was subjected to a single round of black-box testing. In other words, the user testing the application had no prior knowledge or explanation of the application, only a user manual to aid in navigation.

The testing itself was performed on a MacBook Pro. The user was given a set of instructions for tasks to be accomplished while researchers observed and recorded the outcomes. Initially two words were chosen for testing purpose – ‘*Vanakkam’* and ‘*Tamizh’*. The rationale for the choice of these words is on the basis of the presence of the unique syllable ‘*zha*’ in the word ‘Tamizh’ and ‘*Na’* in the word ‘*Vanakkam’*. Upon completion of a working prototype of the application, it was identified that it did not satisfy the conceptual framework of ‘*what you speak is what you get’*. We are not basing our conclusion almost entirely on these two words. However, since the language is syllabic in nature and the one of the main focus within the language were on syllables unique to the language such as ‘*zha’, ‘La’* etc, a couple of words such as ‘*Vanakkam*’ and ‘*Tamizh’* would be sufficient to indicate whether or not the application fulfills the conceptual framework. This, we argue is dependent on the target language and may not be applicable to other languages.

The outcomes of these tasks were measured on a success/failure scale. Four separate test cases were completed during the testing phase with varying tasks for both the application and the user to complete successfully which allowed for testing of the interface design usability and the application functionality at the same time.

5. Results

The user was able to successfully navigate the application to accomplish the primary task of speaking into the microphone and the application converting that to Tamil orthography. Usability was high throughout the majority of the application with a few features, such as the copy and paste button, providing the user with problems.

Specific to application performance the test cases provided mixed results. In Test case one, voice-input data, the application was successful in 100% (4 of 4) of tasks tested and 66.67% (4 of 6) of total tasks. Two tasks in test case one were unable to be completed due to testing environment. Test case two showed the application’s accuracy at 66.67% (2 of 3) of the tasks tested and 50% (2 of 4) of the total tasks. One task of test case two was unable to be tested due to lack of access to API server-side functionality. In test case three the application successfully completed 100% (2 of 2) of the tasks tested and 50% (2 of 4) of the total tasks. Again, the two untested tasked were due to lack of access to server-side API access. Finally, in test case four the application successfully completed 33% (1 of 3) of tasks tested and 25% (1 of 4) of the total tasks. One task was not able to be tested due to testing environment. In total, the application successfully completed 75% (9 of 12) of tasks tested and 47.9% (9 of 19) of the total tasks in the test cases.

6. Discussion

This section discusses the implications of the results and further development of the application. The results of the testing of this application provide mixed feedback. The main functionality of this application was developed successfully using available industry standard tools. The user was able to speak into the application and receive accurate feedback in Tamil orthography. One of the main requirements of this application was not met, though. When the user would pronounce words or syllables incorrectly, the application would not provide the intended result. This application, like many other applications, would predict what it anticipated the user meant to say instead of displaying the result exactly as uttered by the user, even if pronounced incorrectly. Several reasons led to these results such as using a closed source API, development abilities, and development time due to circumstances that arose during the process.

The model proposed in [3] to predict the user acceptance of speech to text application is based on the UTAUT model. Keane (2017) discusses the methodology for decolonisation. [43] also provides an insight on the relationship between identities and script. Arguably, the identities apart from Tamil speaking Muslims, the concept of script and identity could also be related to the Tamil speaking Brahmans [44] who speak a highly Sanskritised Tamil as pointed by [43], [44], [5]. The used of the term ‘Sanskritised Tamil’ suggests presence of Sanskrit vocabulary in Tamil arguably by a minority of the Tamil society. There are compelling reasons to consider the aspect of code mixing, code switching, accuracy of pronunciation, choices of orthography as emphasised in [3]. Code mixing, according to [45] are of three types: insertion or embedding, alternation between the structures, congruent lexicalisation.

On the other hand, Creole of the Indian ocean such as Mauritius presents interesting insight on the value of language, and the kind of transformation, the society undergoes over time. The usefulness of English in the economic front, motivated them to retain it as an official language unlike the Reunion which is typically French [54]. The work of [47] suggest the indigenisation of the West in Asian culture such as in Malaysia and Singapore. More importantly, it discusses, the negotiation of identity which we argue is crucial from the point of user acceptance of a language based technology as it is inextricably linked to the choice of language, orthography literacy and preference as seen in [3]. [48] provides example of Ramanujam who lived in the United States from 1959, wrote primarily in English despite deeply rooted in South Indian Brahman culture.

The work of [44] provides an interesting insight to the choices made by Brahmans, Tamil Brahmans in their case. [49] presents an interesting studies around language, education from the global southern contexts. In their work, they explore the question “Can English function as the national tongue of the once-colonized nations?”. The work of [49] brings out the struggle between the Tamil purists and ‘others’. Mandarin, Tamil and Malay are important identity markers for the Chinese, Indians and Malays within Malaysia [51]. The work of [50] and [53] provide perspectives on Standard Spoken Tamil.

Tamil talk is an application that was designed on the conceptual framework of ‘What you speak is what you get’ [2]. This meant that any incorrect pronunciation by the user would also result in incorrect orthography. The design of Tamil Talk and the expectation for the users to pronounce syllables accurately comes from the Vedic philosophy and practices of teaching and learning. However, anecdotal evidences of contemporary usage of language meant that aspects such as code-mixing, code switching had to be considered. In a language based application like speech to text, the ability to speak the language or in other words proficiency in the target language inevitably becomes a key requirement for the user to use the application. In this case, the proficiency of language cannot be assumed for a variety of reasons.

The prediction of user acceptance of Tamil talk shall be explored on the basis of the following comments [3] shown in table 1.

Table 1. Comments to explore user acceptance of Tamil talk

On pronunciation	On orthography	Empirical link between society, language and use of technology
Language and spelling cannot be changed.	I am comfortable with Roman orthography	Society influences the usage of Tamil.
You have to pronounce the word correctly.	Tamil must be written in Tamil orthography. You either write English in English or Tamil. Roman orthography is unacceptable.	What do I gain by learning Tamil?
People must change their pronunciation	Tamil words in Tamil, English in Roman	What we speak is not at all Tamil! Sangam Tamil is pure Tamil.

The Tamil talk application as developed by [2] does not solve the issue of code mixing and code switching as well as the issue of mispronounced syllables displaying 'correctly'. Code mixing and code switching is a common phenomenon that occurs in a multi-lingual or bi-lingual environment. Anecdotal evidences suggest that bth code-mixing and code-switching are common occurrences in Tamil speech. An example of code-mixing is

Tableல இருக்கு (It's on the table)

An example of code switching is:

So இதுல compulsion வர்ரதில்ல this is how you are conditioned. [3]

In designing an application, the output orthography plays a vital role and based on the indigenous approach, these decisions need to be made by the designers of the application rather than by the users.

The output orthography of a code mixed sentence is rather unpredictable. The result is contrary to the views expressed in [3]. However, if we were to consider the work of [55], Tamil talk might be appealing to a section of Tamil speaking Brahmans who extensively code-mix, code-switch into English. The work in [3] questions the necessity of a speech to text application in Tamil, in the event of extensive code-switching and code-mixing [3].

The user acceptance model is broadly divided into two: language and technology. We argue that, to predict the user acceptance for a language based technology, particularly for applications like Tamil talk which is speech to text, the ability to speak, read and write should be a non negotiable component embedded within the user acceptance model. Language maintenance, opportunity to use the language, perceived usefulness of the language, government policy on language are factors that arguably contribute towards the usage of language in the social sphere.

The attribution of lack of English proficiency is also perceived to be a sign of being 'backward and uncivilised' could perhaps be one of the strong factors in language shift towards English [56]. [57] points that Tamils notably the Brahmans within the Malaysian context have largely shifted to English and retain Tamil only for cultural and religious purpose. Whilst this is also true in other contexts as seen in [58], [59]. Language maintenance in country like Malaysia where it is not official unlike its neighbour Singapore where it is official, has been challenging [60].

But, the issue of language maintenance is not unique to Tamil. [61] presents a case study on the struggle of the Malayalee community in Singapore in maintaining their language. The argument of Non-Tamil Indian communities in Singapore (Malayalees in this case), is the lack of recognition of Non Tamil Indian languages in Singapore combined with numerically minority contributes towards shift in other official languages such as English and Tamil. Since, Malayalam is closely related to Tamil, it could be argued that the Malayalam speakers could learn Tamil as a second language with ease. Second language speakers of Tamil such as the Malayalees of Singapore would have even less motivation to use Tamil in technology.

One should assume that language maintenance in a country or region where it has an official status should relatively be easy. Interestingly, the study by [62] suggest otherwise. In his study, the respondents showed a shift towards English with a belief that the language can replace all domains even in temples where Sanskrit is used. The situation in the Indian state of Tamil Nadu is not encouraging either. It is interesting to observe that the native Tamil speakers struggle to maintain their language even in spaces where there is some form institutional support but at the same time complain about repressive policies in countries like Myanmar. The Burmese of Myanmar also recognized the importance of English and its role in socio-economic upliftment [63]. In fact this seems to have been the trend in many for colonies including African countries [64].

In fact, one could draw parallels between the introduction of English in Africa and Macaulay's proposal of education system in India. In both cases, they were linked with social mobility, progress and language of the intellect. Post-independence many of the African and Asian colonies continued the use of English. Some of them perceive English as a unifying and level playing language that unites various ethnic groups in the country. Could these be the effects of colonialism and language imperialism? Tamils in Myanmar were sandwiched between Burmese and English. The former was often associated with the national identity as in countries like Malaysia, Sri Lanka where Malay and Sinhala respectively are usually perceived by the majority as identity markers of the nationality. As a result, code-mixing by Tamil parents were actively encouraged with a view that it might contribute towards fluency in English.

Although, we cannot generalize, the tendency to switch to English could also be attributed to the colonialism [65]. The studies suggest that the preferential attitude of a native Tamil towards English is not very different in native space. This attitude is problematic because it contributes towards the expectation when designing a speech to text application like Tamil talk. This is because the application was intended to be designed from the perspective of language and philosophy as opposed to the user's usage. The argument of perceived usefulness of the language as a motivation factor to retain in social sphere which subsequently could potentially be the language used in technology merits consideration and further research.

Although, there are debates over the Standard form of Tamil, and in some cases, on the notion of what constitutes Tamil in terms of language. This could be observed in the following comment from [3]:

"*What we speak is not at all Tamil. Sangam Tamil is pure Tamil".*

This provides an empirical link between the individual perception on what is Tamil, how should it be spoken. Whilst this might fit in the narrative of Tamil purist movement, in practice, the attitude is somewhat different. The pure Tamil movement as seen in the work of [48], was initially directed towards any or perhaps all foreign words and sounds including English, Sanskrit. However, the proponents of pure Tamil movement such as Vedhachalam himself wrote frequently in Tamil. Whilst Vedhachalam's writing in English provides a different view to the people who resonate with the idea of pure Tamil movement or atleast with the concept of speaking pure Tamil. In this context, it is compelling to consider some of the genre that a section of Tamil society seem to identify with.

For example, the Manipravalam (Tamil with a disproportionately large amount of Sanskrit vocabulary) predominantly used by the Tamil speaking Brahmans [44]. On the other hand, Araputamil is a form of Tamil that makes use of liberal Arabic vocabulary [43]. Arguably, there are commonalities in both. The preference of genre by a section of Tamil speaking community could arguably be attributed to religion and philosophy [43]. If we were to consider the orthography, there would be considerable differences in expectation between the two groups.

For example, Tamil- Sanskrit code-switching would involve either the Grantha or the Devanagari script, Tamil-Arabic code switching would involve Tamil-Arabic script or Tamil written in Arabic script as in [43]. Neither of these would be acceptable by the purists as [50] seem to suggest.

Moreover, the Tamil language makes use of six borrowed characters called the ‘Grantha’. From a purist point of view, the Grantha would serve no purpose in representing ‘pure Tamil’. In fact, the Grantha script was accommodated to facilitate representation of Sanskrit sounds. [3], suggests that clearly there are different expectations amongst users in terms of how their speech ought to appear in text. For example, one of his participants preferred:

“*Tamil words in Tamil, English in Roman*”.

Whilst, it is possible to employ bi-lingual and multi-lingual speech recognition, it is appearance of the output that is more challenging from the perspective of readability. The choice of orthography could ideally be classified into one or more of the following reasons:

1) Inability to read, write Tamil either by the user or the person receiving the message.
2) Poor readability as a result of code mixing and code-switching.

The choice of script, code-switching and code-mixing could also arguably be attributed to the users religious or philosophical affiliations. For example, the ‘Arabic-Tamil’ was predominant amongst Tamil speaking Muslims while the use of disproportionately large Sanskrit vocabulary in Tamil was predominant amongst the Tamil speaking Brahmans. Tamil, Sanskrit and Arabic are all different languages, with different script and sounds, possibly with some common sounds. Due to the complexities and dilemma it presents on the output script of a speech to text application, we are compelled to consider the view of a purist as seen in [50] from a perspective of application design.

Although there are many dimensions and contexts to purism, we restrict our view to the use of language in the context of designing a speech to text application alone. The work of [44] and [50] suggest that identity and language are intertwined. Further, the ethnography of [58], [59] the work of [60], [64] further suggests different perspectives on language by various social groups within the Tamil society. The priorities of learning Tamil, its relevance and maintenance of language appears to have been challenged by the native speakers. There are different notions on purism or in other words what constitutes pure Tamil. For example,

“*Sri Lankan Tamil is very pure. It has not got polluted. I am coming from Chennai. It is the worst place to speak Tamil”* [59].

There are different views and debates on language competency and the ability to speak pure Tamil amongst the Indian and Sri Lankan Tamils. Although a few claim that Jaffna Tamil is pure, in practice, the following example from [59] suggest otherwise:

“Okay naan deti eluthirukkiren”.

In our view, pure Tamil of the above example should have been:

“*Sei naan thethiyai ezhuthirukkiraen”(okay I have written the date).*

We argue that the perception of pure Tamil in the context of Indian and Sri Lankan Tamil are relative. Both the variety code mix and code-switch perhaps may be a little less in the Sri Lankan variety. From the perspective of the application, colloquial Tamil or Literary Tamil matters very little since the pronunciation of Tamil syllables would be the same. But what matters much is the borrowing of words from non- Tamil sources such as English, Sanskrit or Arabic. The motivation of speech to text application like *Tamil talk* from the point of view of preserving one’s linguistic heritage is slightly deceptive if we were to consider the following example of the extent of code-switching observed in a conversation:

“I will tell you a time when I was compelled to use technology in Tamizh- Never! By default it was English- there wasn't any compulsion. Its like this, you go to a school, and you graduate out, you be a chemist, you be a pharmacist, you be even a civil engineer, English is your bread and butter, you have to follow English. So இதுல compulsion வர்ரதில்ல this is how you are conditioned.” [4]

The above could be seen in parallel to the contexts of Singapore and Myanmar where a similar perception could be observed in terms of the perceived usefulness of the language. The notion of pure language could also be seen in languages like Sinhala. The author points out the ‘*correct*’ pronunciation of /æ/ instead of the modern usage of /a/. In the case of Sinhala (like the Tamil purists), the emphasis was not only on the pronunciation but also on the grammar. A similar attitude can be seen in [55] work, that centres around the *sabhas* (Theatre) culture of Chennai. The work focuses on the Tamil Brahmans who from the [55] point of view assumes and expects the knowledge of Tamil and English to be able to appreciate the subtle *jokes*. The author further observes that the Tamil Brahmans are notorious for their social conservativism and has an appreciation for traditional values and education.

The practice of code-mixing of two languages at a social level could be argued against the norm of linguistic conservatism. Nevertheless, the Brahman Tamil is an interesting variety to study because of its inherent code switching between Tamil and Sanskrit (*Manipravalam)*. However, the perception of Sanskrit-Tamil variety as quintessential Tamil may be problematic. The expectation of the users using the system may vary quite drastically as seen in [3]. Nevertheless, Sanskrit- Tamil code switching (*Manipravalam*), provides an interesting perspective on the script complexity. The work of Ganesan suggest that *Grantha script* accommodates both the Sanskrit and Tamil sounds.

The same could be argued for *ArwiTamil* where the Arabic script is claimed to accommodate both the Arabic and Tamil sounds. However, over a period, attitude towards Grantha has influenced a shift in script to Devanagari for Sanskrit and Tamil script for Tamil. This has contributed increasing illiteracy of the script and has contributed to the perception that Tamil script cannot handle Sanskrit sounds and the need to use Devanagari script for Sanskrit and Tamil script for Tamil. The question of economic value of the language along with the equation on Sanskrit as a ‘*philosophical language’* meant that the Tamil speaking Brahmans need to acquire proficiency in three scripts namely Tamil, Sanskrit and English.

The work of [65], [44] point to the choices of profession and increasing emigration of Tamil Brahmans to English speaking countries like the United States and the work of [56] suggest that Tamil in non native space refer to a collective cultural identity that share similar values as opposed to the language and the necessity to maintain it. Transliteration is common and positively contributes to the issue of script illiteracy. The dictation experiment in [3] provides an insight on the issues of transliteration which could be argued also contributes to mispronunciation or perceived pronunciation variants in guise of dialects and accents. The *ArwiTamil* also has similar challenges in terms of handling both the Arabic and Tamil sounds in script. In both cases, it is arguably difficult to satisfy users with diverse linguistic preference both in terms of speech and script.

Whilst the rationale to move towards ‘pure Tamil’ might seem a sensible choice to solve much of the design issue of a language based application such as speech to text, it might neither promote the usability of application nor the acceptance of application due to perceived language shift and differing notions of what Tamil is to individual speaker. Therefore, for a native Tamil speaker who frequently code-switches or code mixes into English, we assume that *Tamil talk* application would be fairly well received. However, this needs to be further investigated under controlled conditions. Nevertheless, we take the view that the design of Tamil talk, although not fully achieved, could be argued in favour of preserving linguistic heritage of a society or at least facilitate in the education process of preserving one’s linguistic heritage:

“*Language is like water. Learned men are like filters. Impurities from the outside gets mixed up with water. However, before being used for drinking the water is strained and the impurities are filtered out and discarded In language too, coarse and uncivilised usage seep in. Scholars examine and discard them” [66].*

The design of application from a language perspective as opposed to the user’s usage of the language might contradict the traditional approaches to requirements and user acceptance. Nevertheless, the design view merits consideration when a native view is adopted in designing an application. Language, identity and cultural considerations are areas that need to be further explored in the context of user acceptance, requirement elicitation and feasibility study of software engineering. Since these are inextricably linked to each other, careful design considerations and acceptance framework needs to be established. A deeper understanding of how language and culture operate in a given space, particularly in the native space is undoubtedly important to aid the process of requirement elicitation, feasibility and user acceptance.

In this case, in addition to the language and script, there are a number of other areas that contributes to the challenge of design and acceptance such as language shift, language maintenance, illiteracy in script, institutional policies and so on. Although, there are different views on what constitutes a 'pure form of a language', there appears to be a unanimous agreement. For example, the *Sangam Tamil* or Literary Tamil in the case of Tamil as can be seen in the following excerpt:

"*What we speak is not at all Tamil. We take some words from English, some from Sanskrit. If you talk about proper Tamil, then we need to go to the Sangam Tamil'*[4]

7. Conclusion and further work

This paper has explored the potential acceptance of Tamil talk using relevant literature and framework as proposed in [3]. To be able to predict user acceptance of Tamil Talk, more data is needed. Further work would be to widely test Tamil talk with the native Tamil speakers. In order to provide the best outcome in future development of Tamil speech to text applications we believe that a phonetic dictionary would provide the best result for Tamil speakers. Developing a unique back-end, ideally with the use of open-source API or no API, would ideally allow for the application to achieve the concept of "what you speak is what you get". Future direction could include considering the work of [45] and evaluate its suitability to support the conceptual framework discussed in this paper. Once the application is developed to a reasonable level, the work of [67] might be useful from the perspective of quality testing. The framework of [68] might offer some interesting insight in the context of the reliability and could be further explored as the development progresses. Time constraints, along with the knowledge of development were limitations of this research that hindered the development process. However, it was shown that providing an application for "minority languages" like Tamil is more readily available than ever but not as conceptualized.

This project was challenging; a better understanding of the language would be advised in order to fully create a phonetic dictionary for the outcome to be achieved. Users would discover errors in their pronunciation organically with the use of such an application as words would not be "corrected" for incorrect pronunciation. The user acceptance of Tamil talk cannot be generalised as it lacks data. Since the concept of accuracy of pronunciation arises from the Vedic philosophy, testing it with non-native Tamil speakers in particular with people who do not follow Vedic philosophy might offer some interesting insights. Future work would include extending the application so it fulfils the conceptual framework and use the words tested in [3] as test cases by native Tamil speakers who satisfy the usability criteria.

References

[1] Rabiner L.A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, vol. 77, no. 2, 1989, pp. 257–286.

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

[3] Ramachandran R. Predicting user acceptance of Tamil speech to text by native Tamil Brahmans. Doctoral dissertation. Sheffield Hallam University, 2018, 233 p.

[4] Keane E. Tamil. Journal of the International Phonetic Association, vol. 34, issue 1, 2004, pp. 111-116.

[5] Srinivasan A., Srinivasa K., Kannan K., & Narashimhan D. Speech Recognition of the letter "zha" in Tamil Language using HMM. International Journal of Engineering Science and Technology, vol.1, no. 2, 2009, pp 67-72.

[6] Shulman D. Tamil. Harvard University Press, 2016, 416 p.

[7] Srinivasan A. Real time speaker recognition of letter 'zha' in Tamil language. In Proc. of the Fourth International Conference on Computing, Communications and Networking Technologies, 2013, pp. 1-5.

[8] Crystal D. A little book of language (Little Histories). UNSW Press, 2010, 272 p.

[9] Boyd C. Speech Recognition Technology: The Past, Present, and Future. Available at https://medium.com/swlh/the-past-present-and-future-of-speech-recognition-technology-cf13c179aaf, accessed March 12, 2020.

[10] Fu T., Gao S., Wu X. Improving Minority Language Speech Recognition Based on Distinctive Features. Lecture Notes in Computer Science, vol 11266, 2018, pp. 411-420.

[11] Pandharipande R. Minority Matters: Issues in Minority Languages in India. International Journal on Mulitcutltural Socieitis (IJMS), vol 4, no. 2, 2002, pp 213-234.

[12] Steever S. Introduction to the Dravidian Languages. In Steever S. (ed.). The Dravidian Languages. London: Routledge, 1998, pp. 1–39/

[13] Zvelebil K. Companion studies to the history of Tamil literature. Leiden: Brill, 1992, 291 p.

[14] Lo L. Tamil. Ancient Scripts. Available at http://www.ancientscripts.com/tamil.html, accessed December 1, 2019.

[15] Argondizzo P. How Accurate is Google Translate in 2018? Available at https://www.argotrans.com/blog/accurate-google-translate-2018/, accessed March 10, 2020.

[16] Devarajan S. Relationship between Japanese and Dravidian (Tamil). Available at http://japanese-dravidian.blogspot.com/2009/01/relationship-between-japanese-and.html, accessed March 10, 2020.

[17] Deng Y., Li X., Kwan C. et al. Continuous Feature Adaptation for Non-Native Speech Recognition. International Journal of Computer, Information Science and Engineering, vol:1, no:6, 2007, pp. 1701-1708.

[18] Pornpanomchai C., Ngamwongsakoller P., Tangpitaksame P. & Wonvattanakij C. Thai-Speech-to-Text Transformation Using Dictionary-Based Technique. In Proc. of the International Conference on Networks and Information, 2012, pp. 65-69.

[19] Gales M., Young S. The Application of Hidden Markov Models in Speech Recognition. Foundation and Trends in Signal Processing, 2007, pp. 195-304.

[20] Blanken H., de Vries A., Blok H. & Feng L. (eds.) Multimedia Retrieval (Data-Centric Systems and Applications). Springer Science & Business Media, 2007, 390 p.

[21] Kreyssig F. Deep Learning for User Simulation in a Dialogue System. Master Thesis, University of Cambridge, 2018, 48 p.

[22] Stuttle M. A Gaussian Mixture Model Spectral Representation for Speech Recognition. PhD Thesis, Cambridge University Engineering Department, 2003, 163 p.

[23] Duran N. & Battle S. Probabilistic Word Association for Dialogue Act Classification with Recurrent Neural Networks. Communications in Computer and Information Science, vol. 893, 2018, pp. 229-239.

[24] Graves A., Fernandez S., Gomez F. & Schmidhuber J. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In Proc. of the 23rd International Conference on Machine learning, 2006, pp. 369–376.

[25] Yu D., Deng L. Deep Neural Network-Hidden Markov Model Hybrid Systems. In Automatic Speech Recognition. Signals and Communication Technology. Springer, London, 2015, pp. 99-116.

[26] Clarke S. Measuring API Usability. Dr. Dobb's Journal, vol. 29, 2004, pp. S6-S9.

[27] Samudravijaya K. & Barol M. Comparison of Public Domain Software Tools for Speech Recognition. In Proc. of the Workshop on Spoken Language Processing, 2003, pp. 125–131.

[28] Lange P. & Suendermann-Oeft D. Tuning Sphinx to Outperform Google's Speech Recognition API. In Proc. of the Conference on Electronic Speech Signal Processing (ESSV 2014), 2014, pp. 32-41.

[29] Matarneh R., Maksymova S., Lyashenko S., & Belova N. Speech Recognition Systems: A Comparative Review. IOSR Journal of Computer Engineering, vol. 19, issue 5, 2017, pp. 71-79.

[30] Atwood J. Don't Reinvent the Wheel, Unless you Plan on Learning More About Wheels. Available at https://blog.codinghorror.com/dont-reinvent-the-wheel-unless-you-plan-on-learning-more-about-wheels/, accessed March 10, 2020.

[31] Kepuska V. & Bohouta G. Comparing Speech Recognition Systems (Microsoft API, Google API and CMU Sphinx). International Journal of Engineering Research and Application, vol. 7, issue 3, Part-2, 2017, pp. 20-24.

[32] Lardinois F. Google 1launches an improved speech-to-text service for developers. Available at https://techcrunch.com/2018/04/09/google-launches-an-improved-speech-to-text-service-for-developers/, accessed March 10, 2020.

[33] Techseen Bureau. Google Cloud Speech API now supports 30 more languages. Available at https://techseen.com/2017/08/14/google-cloud-speech-api-update/, accessed March 10, 2020.

[34] Novet J. Google says its speech recognition technology now has only an 8% word error rate. Available at

https://venturebeat.com/2015/05/28/google-says-its-speech-recognition-technology-now-has-only-an-8-word-error-rate/, accessed March 10, 2020.

[35] Juang B.H., & Rabiner L.R. Automatic speech recognition – A brief history of the technology development. In Elsevier Encyclopaedia of Language and Linguistics, 2005, pp. 806–819.

[36] Walker W., LamereP., Kwok, P. et al. Sphinx-4: A flexible open source framework for speech recognition. SMLI TR2004-0811, Sun Microsystems Laboratories, 2004. 9 p.

[37] Twiefel J., Baumann T., Heinrich S., & Wermte, S. Improving domain independent cloud-based speech recognition with domain-dependent phonetic post-processing. In Proc. of the 28th AAAI Conference on Artificial Intelligence, 2014, pp. 1-7.

[38] Ashwell T. & Elam J. How accurately can Google Web Speech API recognize and transcribe Japanese L2 English learners' oral production? The JALT CALL Journal, vol. 13, vo.1, 2017, pp 59-76.

[39] Беленко М.В., Балакшин П.В. Сравнительный анализ систем распознавания речи с открытым кодом. Международный научно-исследовательский журнал, вып, 4 (58) часть 4, 2017 г., стр. 13-18. / Belenko M.V. & Balakshin P.V. Comparative Analysis of Speech Recognition Systems with Open Code. International Research Journal, issue: № 04 (58) Part 4, 2017, pp. 13-18 (in Russian).

[40] Kamarudin M.R., Yusof M.A.F.M., & Jaya H.T. Low cost smart home automation via Microsoft speech recognition. International Journal of Engineering & Computer Science, vol. 13, no. 3, 2013, pp. 6-11.

[41] Lacoma T. How to set up speech-to-text in Windows 10. Available at https://www.digitaltrends.com/computing/how-to-set-up-speech-to-text-in-windows-10/, accessed March 10, 2020.

[42] Manaswi N. Deep Learning with Applications Using Python. Apress, 2018, 219 p.

[43] Tschacher T. From script to language: the three identities of 'Arabic-Tamil'. South Asian History and Culture, vol. 9, no. 1, 2018, pp. 16-37.

[44] Fuller C.J., & Narasimhan H. Tamil Brahmans: The making of a middle-class caste. University of Chicago Press, 2014, 288 p.

[45] Зиеп Н.Н., Жданов А.А. Нейроноподобный подход к распознаванию речи. Программирование, том 44, no. 3, 2018 г., стр. 49-62 / Diep N.N. & Zhdanov A.A. Neuron-Like Approach to Speech Recognition. Programming and Computer Software, vol. 44, no. 3, 2018, pp. 170-180.

[46] Saravanan V., Lakshmi S., & Caleon I.S. The debate over literary Tamil versus standard spoken Tamil: What do teachers say? Journal of Language, Identity, and Education, vol. 8, no. 4, 2009, pp. 221-235.

[47] Wee, C. L. The Indigenized West in Asian multicultures: literary-cultural production in Malaysia and Singapore. International Journal of Postcolonial Studies, vol. 10, no. 2, 2008, pp. 188-206.

[48] Ramazani J., & Ramanujan A.K. Metaphor and postcoloniality: the poetry of AK Ramanujan. Contemporary Literature, vol. 39, no. 1, 1998, pp. 27-53.

[49] Sefa Dei G.J., & Asgharzadeh A. Language, education and development: case studies from the Southern contexts. Language and Education, vol. 17, no. 6, 2003, pp. 421-449.

[50] Kailasapathy K. The Tamil purist movement: a re-evaluation. Social scientist, vol. 7, no. 10, 1979, pp. 23-51.

[51] López C.C. Language is the soul of the nation: Language, education, identity, and national unity in Malaysia. Journal of Language, Identity & Education, vol. 13, no. 3, 2014, pp. 217-223.

[52] Schiffman H.F. Standardization or restandardization: The case for "standard" spoken Tamil. Language in Society, vol. 27, no. 3, 1998, pp. 359-385.

[53] SlobodaT. & Waibel A. Dictionary Learning for Spontaneous Speech Recognition. In Proc. of the International Conference on Spoken Language Processing, 1996, pp. 2328-2331.

[54] Lionnet F. Créolité in the Indian Ocean: Two models of cultural diversity. Yale French Studies, no. 82, 1993, pp. 101-112.

[55] Rudisill K. Everyday Flamboyancy in Chennai's "Sabha" Theatre. Asian Theatre Journal, vol. 29, no. 1, 2012, pp. 276-290.

[56] Canagarajah A.S. Language shift and the family: Questions from the Sri Lankan Tamil diaspora. Journal of Sociolinguistics, vol. 12, no. 2, 2008, pp. 143-176.

[57] Ridge B. National language planning and language shifts in Malaysian minority communities: speaking in many tongues. Amsterdam University Press, 2012, 208 p.

[58] Das Neela S. Rewriting the past and reimagining the future: The social life of a Tamil heritage language industry. American ethnologist, vol. 38, no. 4, 2011, pp. 774-789.

[59] Das S.N. Between convergence and divergence: Reformatting language purism in the Montreal Tamil diasporas. Journal of Linguistic Anthropology, vol. 18, no. 1, 2008, pp. 1-23.

[60] Schiffman H.H.F. Malaysian Tamils and Tamil linguistic culture. Language & Communication, vol. 22, no. 2, 2002, pp. 159-169.

[61] Pillai A.D. Language Shift among Singaporean Malayalee Families. Language in India, vol. 9, 2009, pp. 1-25.

[62] Kadakara S. Status of Tamil language in Singapore: An analysis of family domain, Education Research and Perspectives, vol. 42, 2015, pp. 25-64.

[63] Muniandy M.K., Nair, G.K.S.N., Shanmugam et al. Sociolinguistic competence and Malaysian students' English language proficiency. English Language Teaching, vol. 3, no. 3, 2010, pp. 145-151.

[64] Barnes L. English as a global language: An African perspective. Studies in the Languages of Africa, vol. 36, issue 2, 2005, pp. 243-265.

[65] Fuller C.J., & Narasimhan H. Traditional vocations and modern professions among Tamil Brahmans in colonial and post-colonial south India. The Indian Economic & Social History Review, vol. 47, no. 4, 2010, pp. 473-496.

[66] Coperahewa S. Purifying the Sinhala Language: The Hela Movement of Munidasa Cumaratunga (1930s–1940s). Modern Asian Studies, vol. 46, issue 4, 2012 , pp. 857-891

[67] Липаев В.В. Методология верификации и тестирования крупномасштабных программных средств. Программирование, том 29, no. 6, 2003 г., стр. 7-24 / Lipaev V.V. A Methodology of Verification and Testing of Large Software Systems. Programming and Computer Software, vol. 29, no. 6, 2003, pp. 298-309.

[68] Фролов А.М. Гибридный подход к повышению надежности программных систем. Программирование, том 30, no. 1, 2004 г., стр. 25-36 / Frolov A.M. A Hybrid Approach to Enhancing the Reliability of Software. Programming and Computer Software, vol. 30, no. 1, 2004, pp. 18-24.

Information about authors / Информация об авторах

Raj RAMACHANDRAN SUBRAMANIAN, PhD, Lecturer in Computer Science at the Faculty of Environment and Technology. Research interests include feasibility studies, requirement engineering, user acceptance of language based technologies, and Human computer interaction.

EmmaРадж РАМАЧАНДРАН СУБРАМАНЬЯН, кандидат наук, преподаватель компьютерных наук на факультете окружающей среды и технологий. Научные интересы включают анализ реализуемости проектов, разработку требований, принятие пользователями языковых технологий и взаимодействие человека с компьютером.

Emmanuel Kayode Akinshola OGUNSHILE, Ph.D., Senior Lecturer in Computer Science. Research interests include Automated Specification, Verification and Testing of Software and Hardware, Model Based System Engineering, Formal Methods, Object-Oriented Languages and Type Theory.

Эммануэль Кайоде Акиншола ОГУНШИЛЕ, кандидат наук, старший преподаватель компьютерных наук. Область научных интересов: автоматическая спецификация, верификация и тестирование программного и аппаратного обеспечения, системная инженерия на основе моделей, формальные методы, объектно-ориентированные языки и теория типов.