

Exploring a web-based application to convert Tamil and Vietnamese speech to text without the effect of code-switching and code-mixing

Authors

Organisation.

e-mail: {email}@email

Received

Abstract—This paper attempts to develop an application that converts Tamil and Vietnamese speech to text, with a view to encourage usage and indirectly ensure linguistic preservation of a classical language. The application converts spoken Tamil and Vietnamese to text without auto-correction, code-mixing or code-switching. This paper proposed 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 and Vietnamese speakers. The paper further explores similarities and differences in the two contexts.

DOI:

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, e.g., 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, even after decades of work, 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 innovation [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 (e.g., neurons) bound by weight (e.g., synapses which are changed with the individual experience to support the integration of multiple constraints).

This paper is an extended version of [4] and aims to apply the Conceptual Framework [5] to propose a web-based application that converts Tamil and Vietnamese speech to text without the effect of code-mixing and code switching.

The structure of the paper is as follows:

- Section 2 includes the literature review including the introductions to Tamil and Vietnamese languages, fundamentals of speech recognition, and how the speech-to-text applications work as well as their benefits and challenges.
- Sections 3 describes the proposed application.
- Section 4 consists of the testing results for the proposed application.
- Section 5 concludes the paper and provides future research directions.

2. LITERATURE REVIEW

1. *Introduction to Tamil language*

Tamil is one of the Dravidian languages which is spoken mostly in the state of Tamil Nadu and Puducherry where it is the only official language. There are also large Tamil-speaking populations elsewhere, including Sri Lanka, Malaysia, and Singapore, where Tamil has the status of a national and official language. According to Vistawide, there are over 68 million native speakers of Tamil. Tamil is a diglossia language which has two varieties that are used in different conditions [6]. 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 [7].

As far as speech-to-text applications in Tamil is concerned, 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. One of the main reasons for the lack of accuracy is that there is sometimes a nuance in the written sentence that a machine cannot comprehend the same way as people [8].

2. *Introduction to Vietnamese language*

Vietnamese is a language (with over 90 million native speakers) originated in Vietnam, where it is the national and official language. Its vocabulary is significantly influenced by Chinese and French. Due to migrations, Vietnamese speakers are also found in other parts of Southeast Asia, East Asia, North America, Europe and Australia. Similar to other languages in Southeast Asia and East Asia, Vietnamese is considered an analytic language with phonemic tone.

In regards to speech-to-text in Vietnamese language, there are a wide range of available applications such as FPT.AI [9], Vietnamese Voice Typing Keyboard (Google Play Store), and Google Translation. These applications also face the similar challenges that exists in Tamil speech-to-text applications.

3. *Fundamental of speech recognition*

Speech recognition is a multi-level pattern recognition process that analyses and integrates acoustic signals into a hierarchy of sub-word units (e.g., phonemes), terms, phrases, and sentences [10]. 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.

Fig.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 [11]. 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 [11]. 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 [12].

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 [13].

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.

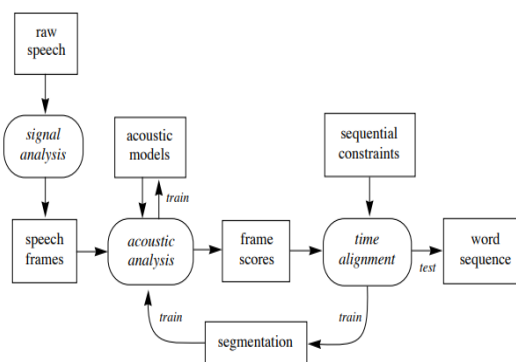


Fig. 1. Structure of a standard speech recognition system [2]

4. How speech-to-text systems 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 could not handle these problems.

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.

5. *Benefits of speech to text*

The benefits of speech to text are as follows.

- **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:** When traditional method is replaced with a mobile transcription app to speak into, one is able to boost their writing 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.

6. *Challenges of speech to text*

Some of the developmental challenges are briefly discussed below:

- **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 [12].
- **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 [12]. In Tamil language, intonation and speech may vary according to geographical region when speaking day to day Tamil, formal Tamil speech does not recognise accents [3]. On the contrary, although

Vietnamese language has three major geographically based dialects namely southern, central, and northern, the speech-to-text tools can still recognise the words (sentences) being spoken.

- **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 [12].
- **Code-switching:** It is when a speaker alters between two or more languages in a sentence or conversation (e.g., Are we eating *chez ta mere demain?* – In this sentence, there are both English and French words). Examples of code switching in the context of Tamil and Vietnamese languages are as follows. In *Tamil*: Book tableவ இருக்க. In *Vietnamese*: Bạn trễ deadline rồi (English meaning: You have been late for the deadline.)
- **Code-mixing:** Code-mixing is the mixing of two languages to form a word. Some examples of code-mixing in Tamil are as follows: classஉக்கு, roadவ. There is no code-mixing in the Vietnamese language.
- **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, in the context of Tamil language, 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. Vietnamese language, on the other hand, is widely spoken by all Vietnamese people as it has been the only and primary language of Vietnam since the late 19th century.

3. THE PROPOSED APPLICATION

1. 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 publicly 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 paper.

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

2. Functional requirements

The requirements in this paper are based on Mann et.al [4] and Ramachandran [5]'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 and Vietnamese orthographies to represent the spoken Tamil and Vietnamese words.
- 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.

Below are the functionalities of the application which are in line with the provided requirement specification document of the client.

a) The application consists configured lists of 28 Tamil words and 14 Vietnamese words (any other words 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 users 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 the recognised Tamil and Vietnamese words to text, respectively.

3. *The developed prototype*

This application was developed by a team of seven members who have diverse experience and skillsets. The following technologies were used for developing the application.

- Java
- MySQL
- Apache Server

- Spring MVC Model
- Hibernate
- JSP
- Servlet

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

The final prototype was executed by allowing Tamil and Vietnamese speakers to speak into the microphone of the system to verify the resulting text and complete the test case.

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 [14]. 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 [14]. The quality control the group is going to adopt is to perform product testing with the intention 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 [14].

4. TESTING

The prototype was due to be tested at each iteration of the development, with live testing carried out by the native Tamil and Vietnamese 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 testing strategy had to be revised as face-to-face testing was no longer possible.

The hardware such as the microphone and speaker were properly checked 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 development, native Tamil and Vietnamese speaking team members

tested the system by speaking pre-selected words in Tamil and Vietnamese into the microphone to verify the resulting texts.

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 and not recognising 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, we decided to create a new application.

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 synthesised 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 or Vietnamese word. The utterance was recorded, and the audio file was sent to our server, which performed the speech recognition task and passed the recognised Tamil or Vietnamese Unicode text to Google convert API. The recognised Tamil or Vietnamese text and its synthesised waveforms were then transferred to the client-side and made available for viewing.

It is important to note that, the system is designed to not display anything given the following scenarios:

- When the user mispronounces the word.
- When the user speaks words other than those in the speech corpus provided by the client.
- When user speaks a language different from Tamil and Vietnamese.

1. Testing against Tamil

The system was tested against a configured list of 28 Tamil words (Table 1). 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, ப 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 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 was not 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 application, 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 in Fig. 2, the Yes or No output indicates whether the system displayed the correct word.

Table 1. Tamil words for testing

Tamil orthography	Roman orthography
அலை	Alai
முகநூல்	Muganool
அலை (அழை)	Alai (Azhai)
அளைப்பிதல் (அழைப்பிதழ்)	ALaippithal (azhaippithazh)
விநூப்புரம் (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	EtrukkoL
எனதுகோள் (எழுதுகோல்)	ELuthugol (Ezhuthugol)
மனப்பான்மை	Manappanmai

	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	அனைப்பிதல் (அ	ALaippithal	Yes	YES	NO
6	வினாப்பரம் (விழ	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

Fig. 2. Tamil test results

To examine the difference between the proposed 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 showed that the text 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.

2. Testing against Vietnamese

The system was tested against a list of 14 Vietnamese words (Table 2) that tend to be mispronounced by the native speakers. Similar to Tamil testing, 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.

Again, the system was tested by collecting pre-recorded audio from a native Vietnamese speaking team member and those recordings were used for testing the application. As showed

in Table 3, the Yes or No output indicates whether the system displayed the correct word.

Table 2. Vietnamese words for testing

Vietnamese orthography	Roman orthography
Chín chu	Chin chu (chinh chu)
Chua xốt	Chua xot (chua sot)
Cọ sát	Co sat (co xat)
Giành giật	Gianh giat (danh giat)
Khất khe	Khat khe (khac khe)
Lãng mạn	Lang man (Lang mang)
Mãi mê	Mãi mê (mãi mê)
Muối mẫn	Muoi man (mui man)
Xán lạn	Xan lan (sang lang)
Se sua	Se sua (xe sua)
Văng cảnh	Văng cảnh (văn cảnh)
Xạo sự	Xao su (xao xu)
Súc tích	Suc tich (Xuc tich)
Xoay sở	Xoay so (Xoay xo)

Table 2. Vietnamese test results

Vietnamese orthography	Roman orthography	Khoa (tester)
Chín chu	Chin chu (chinh chu)	NO
Chua xốt	Chua xot (chua sot)	YES

Cọ sát	Co sat (co xat)	NO
Giành giật	Gianh giat (danh giat)	YES
Khất khe	Khat khe (khac khe)	YES
Lãng mạn	Lang man (Lang mang)	YES
Mãi mê	Mãi mê (mãi mê)	YES and NO → The system displayed both correct and incorrect words.
Muối mẫn	Muoi man (mui man)	NO
Xán lạn	Xan lan (sang lang)	NO
Se sua	Se sua (xe sua)	YES
Văng cảnh	Văng cảnh (văn cảnh)	YES
Xạo sự	Xao su (xao xu)	YES
Súc tích	Suc tich (Xuc tich)	NO
Xoay sở	Xoay so (Xoay xo)	YES

We also evaluated these Vietnamese words against Google speech-to-text application and interestingly, it showed a consistent result with our proposed application. Both the systems had difficulties in identifying the word “mãi mê” when in some tests, they displayed the correctly pronounced word and in some other tests, they displayed the mispronounced word.

3. Testing against Vietnamese

This paper proposed a web-based speech-to-text application that enables users to convert spoken Tamil

and Vietnamese into texts. 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 and Vietnamese speakers. In essence, the users should be to speak a number of Tamil and Vietnamese words from the pre-defined pools of words and receive accurate text in Tamil and Vietnamese orthographies. However, the application could not produce any output when incorrect pronunciation of the word in question occurred. This continues to be a major area of research and focus.

The research was conducted under the Covid-19 pandemic which provided us a unique opportunity to explore alternate methods 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.

5. CONCLUSIONS

This paper proposed a speech-to-text application using Google's cloud conversion API, which can be used to convert spoken Tamil and Vietnamese speech to text. This web-based application is useful for users who wish to learn and practice the proper pronunciation of Tamil and Vietnamese words. The team has built ASR and used Google's speech to text translation tool to build this application. This paper enables Tamil and Vietnamese languages to be spread and recognised 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 work also emphasises the indigenous 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 recognise and convert any word in Tamil and Vietnamese, instead of the limited number of words specified in section 4. We will integrate an error feedback system which notifies the users when any code-mixing or code-switching is detected, and if any words from a different language are spoken into the application.

REFERENCES

- [1] N. Fearn and B. Turner, "Best speech to text software in 2020: Free, paid and online voice recognition apps and services," 18 04 2020. [Online].
- [2] J. Tebelskis, "Speech Recognition using Neural Networks," 1995.
- [3] K. Sakai, "Language acquisition and brain development," *Science*, vol. 310, no. 5749, pp. 815-819, 2005.
- [4] D. Mann, N. Weston, K. Frederic, E. Ogunshile and R. Ramachandran, "Tamil talk: What you speak is what you get!," *2019 7th International Conference in Software Engineering Research and Innovation (CONISOFT)*, pp. 36-48, 2019.
- [5] R. Ramachandran, "Predicting user acceptance of Tamil speech to text by native Tamil Brahmins," *Dotoral dissertation*, 2018.
- [6] C. Ferguson, "Diglossia," *word*, vol. 15, no. 2, pp. 325-340, 1959.
- [7] E. Keane, "Tamil," *Journal of the Internation Phonetic Association*, vol. 34, no. 1, pp. 111-116, 2004.
- [8] N. McGuire, "How Accurate is Google Translate?," 2018. [Online]. Available: <https://www.argotrans.com/blog/accurate-google-translate-2018/>. [Accessed 18 03 2020].
- [9] FPT.AI, "Speech to text," 2021. [Online]. Available: <https://fpt.ai/stt>. [Accessed 14 04 2021].
- [10] J. Orchilles, "Speech Recognition," 2010. [Online]. Available: <https://www.sciencedirect.com/topics/computer-science/speech-recognition>. [Accessed 29 01 2020].
- [11] V. Prasad, "Voice recognition system: Speech-to-text," *Journal of Applied and Fundamental Sciences*, vol. 1, no. 2, p. 191, 2015.
- [12] Stephanie, "The current challenges of speech recognition," 2019. [Online]. Available: <https://onlim.com/en/the-current-challenges-of-speech-recognition/>. [Accessed 29 01 2020].

[13] A. Madhavaraj and A. Ramakrishnan, "Design and development of a large vocabulary, continuous speech recognition system for Tamil," *2017 14th IEEE India Council International Conference (INDICON)*, pp. 1-5, 2017.

[14] A. Way, "Quality expectations of machine translation," *Translation quality assessment*, pp. 159-178, 2018.