Preserving Linguistic Heritage for Generations to Come!

Adarsh Appannagari, Manideep Chittineni, Sathvik Jetty, Zinnia Sarkar,
Vinod Eslavath, Raj Ramachandran, Emmanuel Ogunshile

University of the West of England, Bristol, United Kingdom
raj.ramachandran@uwe.ac.uk

Abstract. Speech to Text is the ability to convert spoken word to text. There are many speech to text conversion applications available for different languages. Tamil is an ancient classical language which is vastly spoken in southern parts of India, Sri Lanka, Malaysia, and Singapore. The speech to text application is designed from an indigenous perspective to enable the native speakers to preserve their linguistic heritage. The application was developed using agile methodology and the testing of the application suggested that the existing API do not support in the notion of preserving the language in its original form.

Keywords: Linguistic Heritage, Tamil, Speech to Text, Indigenous Method, Software Engineering.

1 Introduction

Tamil is a syllabic language with almost one to one correspondence between a syllable and orthography. It is one of the longest surviving classical language and civilizations of the world with over 2000 years of antiquity. It was one of the first Indian languages to be accorded the 'classical' status in 2004 having fulfilled four key criteria set by the government. The key criteria include "high antiquity over a period of 1500-2000 years, body of ancient literature/ text which is considered as valuable by generations of its speakers, literary tradition be original and not borrowed from another speech community and finally classical language and literature being distinct from modern"(PIB, 2020). The criteria of original literary tradition and not borrowed from other speech community is noteworthy because Tamil is considered to be one of the most conservative languages of India with extremely limited borrowings from other languages notably Sanskrit. The work of Kailasapathy (1979) explores the various facets of the Tamil community, linguistic purism in particular. Among the many unique aspects of the language, the syllable '*zha*' is considered special and unique. Infact, the original pronunciation of the language contains this syllable (Tamizh). Shulman (2016) reiterates that it is best for others to listen to a native Tamil speaker uttering the 'zha' syllable. However, Ramachandran (2018), McDonough, J., & Johnson, K. (1997) among few others have identified that not all native Tamil speakers are able to quite accurately pronounce the syllable 'zha'. The work of Ramachandran (2018) focuses on three key aspects namely: code-mixing and code switching, accuracy of pronunciation and orthography.

Digital Presentation and Preservation of Cultural and Scientific Heritage. Conference Proceedings. Vol. 10, Sofia, Bulgaria: Institute of Mathematics and Informatics – BAS, 2020. ISSN: 1314-4006, eISSN: 2535-0366

In the context of speech to text, these become increasingly relevant and important. Usually, a software or application is developed based on the users requirements. Considering the nature of the language and the emphasis on accuracy of pronunciation from a philosophical point of view, Ramachandran (2018) proposed an indigenous approach to design, development of language based technology as well as a user acceptance framework. Mann et.al (2019) attempted to develop an application on the conceptual framework 'what you speak is what you get' but the core objectives were unaccomplished. Whilst there are existing speech to text applications in Tamil, the rationale for attempting to develop an application is based on Ramachandran (2018) work which would enable native Tamil speakers to preserve their unique set of syllables that would largely involve oral tradition of teaching and learning broadly encompassing accuracy of pronunciation, conformance to language syntax as well as script. In this paper, we explore the feasibility of developing a speech to text application that would enable the native Tamil speakers to preserve their linguistic heritage.

Speech is the basic, common, and efficient form of communication for people to interact with each other. Now a day's speech to text technologies are commonly available for a finite but with an interesting range of task. This technology enables machines to listen human and respond correctly to human speech and provide accurate result (Lee, Glass, Lee, & Chan, 2015). In the current scenario of evolving technologies (Google Home, Alexa, Siri etc.) humans prefer communication over speech rather than providing inputs via typing on a keyboard, since the Communication among human being is dominated by speech, thereby it is natural to expect speech interfaces with computer (Zhihong, Pantic, Roisman, & Huang, 2009).

This can be accomplished by developing speech recognition system: speech-to-text which allows computer to translate speech request and dictation into text. (Adetunmbi, Obe, & Iyanda, 2016) Speech recognition system: speech-to-text is the process of converting an audio signal to a set of words which is captured by a microphone of a device.

1.1 Pronunciations

The speech to text recognition engine employees all sorts of information, statistical models, and calculations to change over speech input into content (Huang, Liu, Shadiev, Shen, & Hwang, 2014). Speech to text engine receives data and articulates the input and provides the output (Virtanen, 2013). Words can have multiple pronunciations associated with them. For example, the word "the" has at least two pronunciations in the U.S. English language: "thee" and "thuh." (The American Heritage guide to contemporary usage and style, 2005). In Tamil language there would be no such problems as Tamil language has a one to one pronunciation.

The vocabulary size of speech recognition system affects the processing requirements, accuracy, and complexity of the system. In speech recognition system, types of vocabularies can be classified as follows:

1. Small vocabulary: single letter.
2. Medium vocabulary: two or three letter words.
3. Large vocabulary: more letter words.

In this paper we focus on medium vocabulary (two or three letter words).

1.2 Working of Speech to Text

The speech recognition engine has a complex task to handle, which takes raw audio input and translates it to recognized text that an application understands (Stinson, Elliot, & Kelly, 2017). The major phases are:

1. Audio input
2. Grammar
3. Acoustic Model
4. Recognised text

Input signal- speech input by the user.

Feature Extraction. It retains useful information of the signal, deduct redundant and unwanted information, show less variation from one speaking environment to another, occur normally and naturally in speech.

Acoustic model. It contains statistical representations of each distinct sounds that makes up a word.

Decoder. It decodes the input signal after feature extraction and will show the desired output.

Language model. It assigns a probability to a sequence of words by means of a probability *distribution*.

Output. Interpreted text is given by the system.

The speech recognition engine converts spoken input into text. To do this, it employs all sorts of data, statistics, and software algorithms. The engine searches for the best match only after the speech data is in the proper format. It does this by taking into consideration the words and phrases it knows about (the active grammars), along with its knowledge of the environment in which it is operating. The information of the environment is given within the shape of an acoustic model. Once it identifies the most likely match for what was said, it returns what it recognized as a text string. (Dawson, et al., 2014)

Most speech engines find a match and are usually very "forgiving." But it is important to note that the engine always returns its best guess for what was uttered. (Virtanen, 2013)

2 Literature Review

An API or application Programming Interface is a software that allows two applications or two machines to communicate to each other. To put it in a more simplistic way, an API serves as a messenger that sends requests to the provider of the service and returns

a response to the requests (Subramanian & Pethuru, 2019). It specifies components that are independent of their respective implementations in a way that they can still vary in implementation definition and programming language without compromising their co-ordination of task. It could be treated as a set of routine protocols used by developers for building a software application. They have certain level of abstraction from the developers who are using it so how they exactly operate can vary from one API to another but still be unknown to their user (Ohnishi, et al., 2019). They define the ways in which different software components should interact.

The top 5 speech to text APIs now that are doing well in the global market are as follows (Simpson, 2019):

1. Google API
2. Microsoft's Voice Recognition API
3. Dialog flow API.AI
4. IBM Watson Speech to Text API
5. Speech - matics API

We have focused on the Google API as this is the basis of our project implementation for speech to text conversion in Tamil. Google API's are more efficient and accurate, hence the rationale for Google API over other API'S from the market (Këpuska & Bohouta, 2017).

Conversion of speech to text can be viewed as a speech recognition service enabling real time transcription of audio streams (Rustam, Huang, Shing, & Hwang, 2014). The Google API recognizes more than 120 languages and varieties in dialect for global support The Google API for Speech to Text implements synchronous recognition request for recognizing speech from audio data input. It can process unto 1 minute of speech audio data sent in a synchronous request. Once the processing and recognition of the audio data is over it sends back a response (Choi, Gill, Ou, Song, & Lee, 2018).

3 Project Goals

The main aim of the project is to explore the feasibility of a speech to text web-based application using the conceptual framework of *what you speak is what you get*.

1. To understand the structure of the Tamil language, background of native Tamil speakers.
2. To evaluate the feasibility and applicability of various speech recognition and speech to text models and techniques that have been used in other languages in this context.
3. Create a web-based application that recognizes a Tamil word, converts it into Tamil orthography as spoken by the user.
4. Create a speech corpus with 28 words mentioned as in Ramachandran (2018)

3.1 Product Use Cases

1. A native Tamil speaker with correct pronunciation.
2. Tamil speech without a code-switched utterance (Example: Tamil - English)
3. Speech to text in the context sending an email or text message on a

smartphone.

The application is designed, developed and tested with the words as seen in (Ramachandran, 2018)

4 Testing and Verification

One of the important things of a software development life cycle is testing. Utmost importance to the testing were given to ensure the quality of the application. Testing in different phases of the development cycle were carried out:

1. Design
2. Coding

The application was tested on the following parameters:

1. Functionality:
   Basic functionality of the UI and speech engine is done by black box testers.
2. Scalability:
   The web-based application was tested to scale the capability of the application e.g.: Knowing the accuracy of the speech to text engine.
3. Reliability:
   The ability of web application is tested with given environmental conditions such as background noise, incorrect pronunciation etc.
4. Usability:
   User Interface of the system.

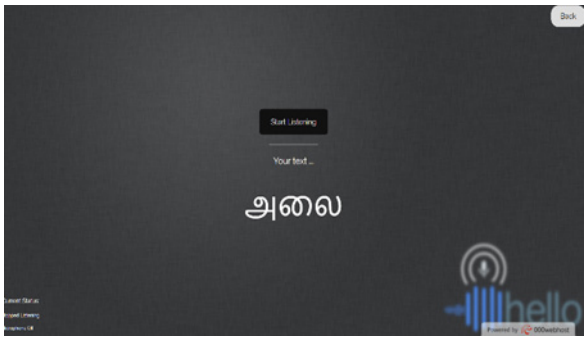
Testing has been carried out rigorously in every stage of the development. This is been carried out by purely black-box testers who does not have any idea about the application or the language.

The rationale for choosing a member of the team who do not speak Tamil language is, they wouldn't know the right pronunciation of the word there by preventing the search engine from making any assumptions over wrongly spelt words and give wrong output. In doing this we would know where to improve the most and work on the engine.

4.1 System Testing

System testing was carried out entirely by black box testers. The expectation from the system is to display the output in the spelling that corresponds to the pronunciation and *not* the correct spelling. In doing so, the user is expected by virtue of the design to either accurately pronounce the syllables or accept the spelling that is in accordance to the user's pronunciation even if it is incorrect.

Tamil orthography: அலை
Roman orthography: Alai
Expected Result: As expected (See below)
Pass/Fail: Pass



The results quite clearly demonstrate that the application does not fulfill the requirements. The main functionality which was to convert speech to text in Tamil was developed appropriately using the Web-Speech API. The user was able to speak and check the output in Tamil orthography. However, the exact requirements remains unfulfilled. The application returns the correct word for incorrect pronunciation and in some cases fails to distinguish between the more nuanced pronunciation such as ற, ன and ண. We predict that this could have been easy to overcome by using a speech dictionary.

5 Conclusion and Further Work

This paper discussed about developing Speech to Text in Tamil Language proposed by Ramachandran (2018). We argue that adopting a conceptual framework that is deeply committed to preserve the linguistic heritage of a society would be of huge benefit to the endangered languages and linguistic traditions that boasts significant antiquity. This would also mean that unlike the technology adapting to the human speech, the users would need to adapt to the design of the system. This would mean, that the design of the application would primarily compel the user to *accurately pronounce syllables and avoid code-switching and code*-mixing. We recognize that the design techniques and user acceptance framework applied for Tamil may need to be adapted for other languages and cultures. Nevertheless, this is a niche area that needs further research. We have proposed an indigenous approach to design and development of a language based technology such as speech to text. Empirical evidence, testing results and from the work of Mann et.al (2019) suggest that existing APIs are less useful in achieving the objective. Future work could involve creating a speech corpus and using a dictionary method in developing a speech to text application. Existing tools could be used to create a phonetic dictionary if the phonemes are ready. We also recommend having a native Tamil speaker in the team who is fluent with Tamil language to help with correct pronunciation of the language, which would be helpful while creating a phonetic dictionary. We take a futuristic view that in a multicultural world, that is rapidly evolving, technology should take the lead in enabling all societies to preserve their rich linguistic heritage for generations to come.

References

Adetunmbi, O., Obe, O., & Iyanda, J. (2016). Development of Standard Yorùbá speech-to-text system using HTK. *International Journal of Speech Technology, 19*(4), 929-944. doi:10.1007/s10772-016-9380-2

Choi, J., Gill, H., Ou, S., Song, Y., & Lee, J. (2018). Design of Voice to Text Conversion and Management Program Based on Google Cloud Speech API. *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*. doi:10.1109/csci46756.2018.00286

Dawson, L., Johnson, M., Suominen, H., Basilakis, J., Sanchez, P., Estival, D., . . . Hanlen, L. (2014). A usability framework for speech recognition technologies in clinical handover: A pre-implementation study. *Journal of Medical Systems, 38*(6). doi:10.1007/s10916-014-0056-7

Huang, Y.-M., Liu, C.-J., Shadiev, R., Shen, M.-H., & Hwang, W.-Y. (2014). Investigating an application of speech-to-text recognition: a study on visual attention and learning behaviour. *Journal of Computer Assisted Learning, 31*(6), 529-545. doi:10.1111/jcal.12093

Kailasapathy, K. (1979). The Tamil purist movement: a re-evaluation. *Social scientist*, 23-51

Këpuska, V., & Bohouta, G. (2017). Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx). *International Journal of Engineering Research and Applications, 7*(3), 20-24. doi:10.9790/9622-0703022024

Lee, L.-S., Glass, J., Lee, H.-y., & Chan, C.-a. (2015). Spoken Content Retrieval—Beyond Cascading Speech Recognition with Text Retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 23*(9), 1389-1420. doi:10.1109/taslp.2015.2438543

Mann, D., Frederic, K. Weston, N., Ramachandran, R. & Ogunshile, E (2019). *Tamil*

/McDonough, J., & Johnson, K. (1997). Tamil liquids: An investigation into the basis of the contrast among five liquids in a dialect of Tamil. *Journal of the International Phonetic Association*, *27*(1-2), 1-26.

Ohnishi, Y., Yamaguchi, S., Shimoikura, Y., Nishino, K., Kondo, H., & Hayashi, A. (2019). Prototype Design of Playback and Search System for Lecture Video Content using Google Cloud API. *Procedia Computer Science, 159*, 1517-1526. doi:10.1016/j.procs.2019.09.322

Rustam, S., Huang, Y.-M., Shing, N., & Hwang, W.-Y. (2014). Review of Speech-to-Text Recognition Technology for Enhancing Learning. *Journal of Educational Technology & Society, 17*(4), 65-84. Retrieved April 26, 2020

Ramachandran, R., 2018. Predicting user acceptance of Tamil speech to text by native Tamil Brahmans (Doctoral dissertation, Sheffield Hallam University).

Shulman, D. (2016). *Tamil*. Harvard University Press.

Simpson, J. (2019). 5 Best Speech-to-Text APIs | Nordic APIs |. Retrieved April 20, 2020, from https://nordicapis.com/5-best-speech-to-text-apis/

Stinson, M., Elliot, L., & Kelly, R. (2017). Deaf and Hard-Of-Hearing High School and College Students' Perceptions of Speech-To-Text and Interpreting/Note Taking

Services and Motivation. *Journal of Developmental and Physical Disabilities, 29*(1), 131-152. doi:10.1007/s10882-017-9534-4

Subramanian, H., & Pethuru, R. (2019). *Hands-On RESTful API Design Patterns and Best Practices: Design, develop, and deploy highly adaptable, scalable, and secure RESTful web APIs.* Birmingham, UK: Packt Publishing. Retrieved April 20, 2020

The American Heritage guide to contemporary usage and style (1st ed.). (2005). Boston: Houghton Mifflin. Retrieved April 04, 2020

Virtanen, T. (2013). *Techniques for noise robustness in automatic speech recognition.* Chichester: Wiley-Blackwell. Retrieved April 04, 2020

Zhihong, Z., Pantic, M., Roisman, G., & Huang, T. (2009). A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 31*(1), 39-58. doi:10.1109/tpami.2008.52

Received: June 05, 2020
Reviewed: June 20, 2020
Finally Accepted: July 12, 2020