Kelvi: A Morphological Parser to Support Tamil Literacy

Shankhalika Srikanth[1] shankhalika.srikanth@alumni.utoronto.ca
Sabrina Yu[1] sab.yu@alumni.utoronto.ca
Sophia Chan[1] ceasings@gmail.com
Madeline Solis de Ovando[1] madelinesdeo@gmail.com

[1]Independent Researcher

Abstract

In this paper we discuss the development of kelvi.ca[1], an open source web-based dictionary and morphological parser designed to aid Tamil learners in developing their literacy skills. Tamil is an agglutinative language and heavily suffixal. Existing Tamil dictionaries only carry stems, not conjugated or inflected forms, and for a beginner learner of the language, isolating the stem in an unfamiliar word can be very challenging. Kelvi provides 1) the stem of any input word alongside its definition, and 2) non-technical descriptions of any suffixes that are part of this input, so that learners will gradually start to recognize these suffixes and be able to understand and produce new Tamil words themselves. In detailing our process of collaborative research, user interviews, suffix database creation, and error analysis, we also hope to show that Kelvi can be adapted for other languages and has the potential to be a useful pedagogical aid for learner literacy development, especially for agglutinative and/or polysynthetic languages which tend to be otherwise underserved in the mainstream.

1 Introduction

Tamil is a Dravidian language spoken in South and Southeast Asia. With its largest speaker population in southern India (73.4 million speakers), it has official language status in the state of Tamil Nadu and the union territory of Puducherry, as well as in Sri Lanka (4.2 million), and Singapore (688.6 thousand). Tamil is also widely spoken in Malaysia (4.8 million) and South Africa (600 thousand). In addition, there are significant diaspora populations in Europe, North America, and Australia; Tamil speakers globally number well over 80 million (World Population Review, 2026). The diaspora population in Canada largely consists of heritage speakers - that is to say, people who grow up hearing their family speak Tamil, but aren't necessarily entirely fluent themselves (Polinsky and Kagan, 2007). Two of the main complicating features of Tamil literacy are the language's heavily diglossic nature and its agglutinative properties. Tamil is heavily suffixal, with most grammatical information expressed either as nominal or verbal suffixes. See Figure 1, where a complete utterance is expressed as one word.

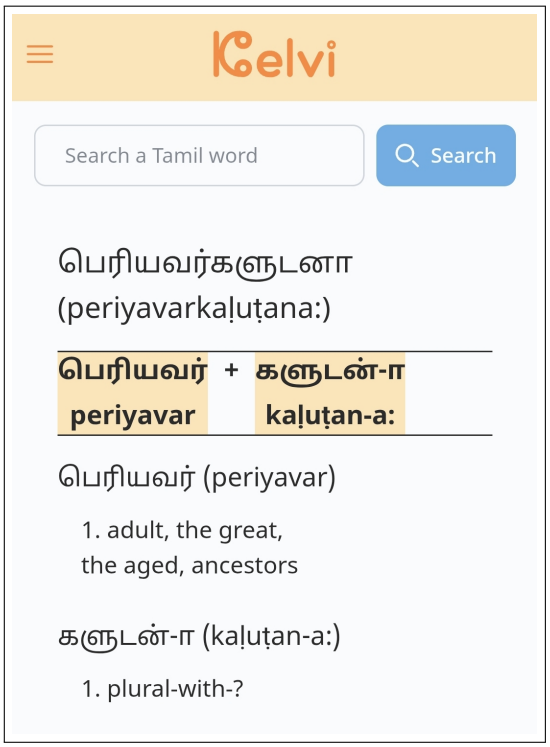

Figure 1: Tamil word as it appears in Kelvi that translates to the question "with the elders/adults/ancestors?"

In written Tamil the higher, formal register is used, where every suffix is phonologically expressed in its entirety. In the colloquial spoken register, however, Tamil words undergo massive phonological reduction, so that many sounds

[1]Github repositories for the front-end and back-end are accessible at https://github.com/saltyseadawg/kelvi-frontend and https://github.com/saltyseadawg/kelvi-backend respectively.

Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications, pages 281–291

that are fully expressed in the written form of a word aren't actually pronounced (Annamalai, 2019; Renganathan, 2019). This means that heritage speakers often struggle to recognize the written version of words they have only ever heard in the colloquial spoken register, since the written form "sounds" different from what they're used to (Renganathan, 2019). Conjugated words don't appear in standard dictionaries, only their lemmas do (consider how "go" can be found in an English dictionary but "went" wouldn't be). Learners then face the "dictionary problem"; to summarize Littell et al. (2017), dictionaries are intended to be a tool for language learners, but paradoxically require knowledge of the language to use; in this case, knowledge of what the "dictionary form" or lemma of an unfamiliar word is. To address this issue we built Kelvi, a web-based dictionary and morphological parser that helps a learner who comes across an intimidatingly long Tamil word see how the word is broken down, and what each "part" (morpheme) means in layman terms. The hope is that they come away with a fuller understanding of the word while also starting to recognize Tamil morphological patterns themselves.

2 Literature Review

The one instance known to these authors of a dictionary specifically designed for a heritage/minoritized language that also provides pedagogical information about morphology is the ongoing work on the digital Wendat Dictionary (Lukaniec and Holmes, 2023). More broadly speaking, there are several open-source technologies that have been developed to support language revitalization and maintenance efforts (Brinklow et al., 2020), such as the dictionary-builder Mother Tongues (Littell et al., 2017) and the morphological generator Gramble (Littell et al., 2024) that was used in pedagogical contexts for Oneida (Lu et al., 2024), SENĆOŦEN (Saanich), and nêhiyawêwin (Plains Cree) (Pine et al., 2025a). Effective morphological analysers using FSTs have also been built for similarily agglutinative languages like Turkish (Yıldız et al., 2019). For Tamil specifically there exist tools such as a lemmatizer (Qi et al., 2020), a morphological analyser (Sarveswaran et al., 2021), and a stemmer (Rajalingam, 2025), as well as some pedagogical resources like a dictionary of Tamil verbs (Schiffman et al., 2009) and a few isolated courses/course materials (Annamalai and Asher, 2002; Prasad, 2025; Cheran, 2001); but to the authors' knowledge there is no other free-to-use pedagogical resource which employs a Tamil morphological analyser.

3 Methods

In accordance with principles behind collaborative community based research (Bucholtz, 2021) and user-centered design methods (Wallisch et al., 2019), we involved target users early on in our explorations of the problem space to ensure we were solving a real issue faced by Tamil learners and providing a solution that learners would want to adopt. We wanted to understand what bottlenecks users were currently facing when searching for Tamil word definitions as well as existing pain points they were experiencing.

3.1 Prototype Design

We built a medium fidelity prototype in Figma, an interface design tool, to test our design (accessible here). We wanted to know whether (a) users would be able to determine the meaning of a word when presented with the morphological breakdown; (b) users found the morphological breakdown useful; and (c) users would use the tool if it existed.

3.2 Participants

To determine our design requirements, we interviewed 10 Tamil adult speakers over the age of 20 who were asked to self describe their Tamil language abilities in terms of oral fluency and literacy. Of our participants, three were native speakers, six were heritage speakers, and one was a beginner L2 learner with no Tamil background who had only recently begun learning Tamil. Importantly, oral fluency and literacy were not correlated for our speakers. In other words, speakers that were fluent in Tamil were not necessarily strong readers, such as in the case of SM who was a fluent, native speaker of Tamil with weak literacy skills. As shown in Figure 2, most of the participants had stronger oral abilities compared to reading and writing.

3.3 Exploratory Interview

At the beginning of the session, participants were asked to self-describe their Tamil oral and literacy abilities. In addition, since most of the participants were heritage speakers, they were also asked demographic and language related questions to gain insight into their Tamil proficiency. As described by Montrul (2016), background information, such

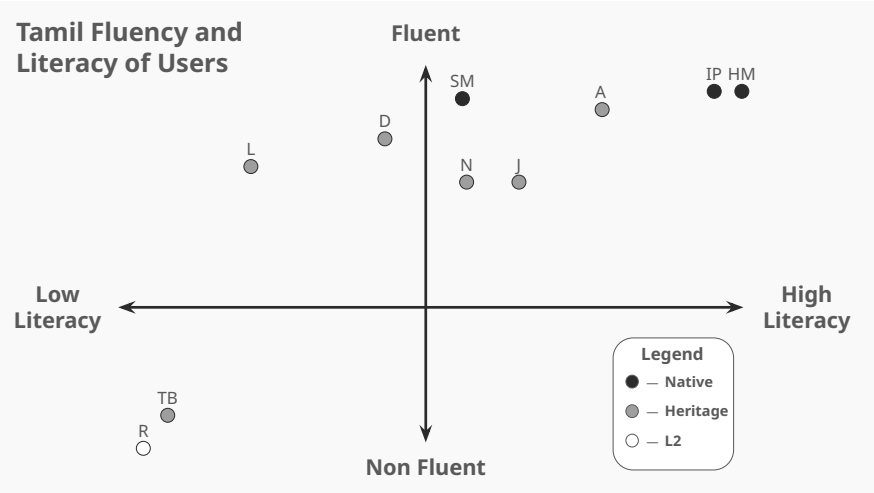

Figure 2: The X-axis represents fluency while the Y-axis represents literacy, forming four quadrants. 6/10 participants fall in the <fluent, high literacy> quadrant, 2/10 participants fall in the <low literacy, non fluent> quadrant, and 2/10 participants fall in the <low literacy, fluent> quadrant. Oral fluency and literacy were thus not correlated among our participants.

as level of education, language of instruction, domains of use of their languages, etc., is important to understanding a heritage speaker's ability in their heritage language. Although the answers to these questions were mostly used to corroborate the responder's self description of their Tamil abilities and not used for any formal analyses, the questions themselves have been included in the Appendix for future researchers. Users were then asked to complete four tasks to test the usability of the prototype's UI as well as input methods, which can also be found in the appendices. After completion of the tasks, users were asked the following questions:

- What do you like/dislike about this tool?
- What features do you wish existed?
- How did each input method compare?
- How did the prototype compare to how you were originally looking up Tamil word definitions in the first task?
- Would you use this tool?

3.4 Interview Results

During the exploratory interviews, participants were asked what tools they were currently using to look up the meaning of Tamil words. The most common answer given was Google Translate, however not all participants were using it just to get the translation of a word. For example, L, who had strong oral abilities but could not read Tamil easily, used Google Translate's TTS feature to hear the word and would then be able to figure out what it meant. On the other hand, IP would use Google Translate to get a Tamil word's romanization which they would then input into the Google search engine to try to find different contexts and uses of the word. For the participants who did rely on the translations provided by Google Translate, several commented that Google's translations were not always trustworthy. Interestingly, none of our participants mentioned using generative AI tools, however this could be due to the fact that most of our participants were over the age of 25. As Magotra et al. (2016) notes, younger individuals are more likely to adopt new technologies compared to older individuals.

When asked if they would use this tool, seven participants responded with yes, two responded with no, and one with maybe. SM, who is fluent with weak literacy skills, stated they would not use the tool. TB, who is a beginner heritage speaker that cannot read, stated the reason behind their "maybe" response was because they felt that their level of Tamil was too low to use the tool, but would use it if it included more conversational words. IP, who is fluent with high literacy skills, felt the tool was unnecessary and that the interface made it hard to correlate meaning with the word parts.

Conversely, the other participants felt that the tool was quite useful. In N's case, they thought that learners would find the pedagogical breakdown

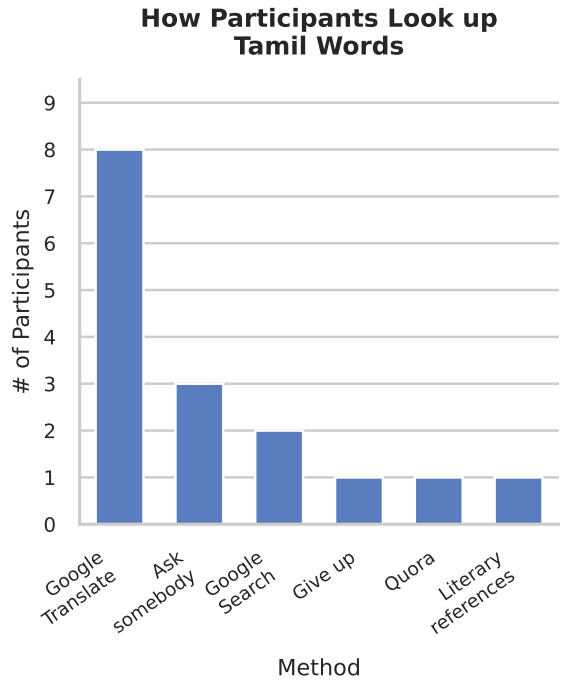

Figure 3: Graph of participants' methods for looking up Tamil words. Google Translate is the most common at 8 votes, while the five other remaining choices have 3 votes or less.

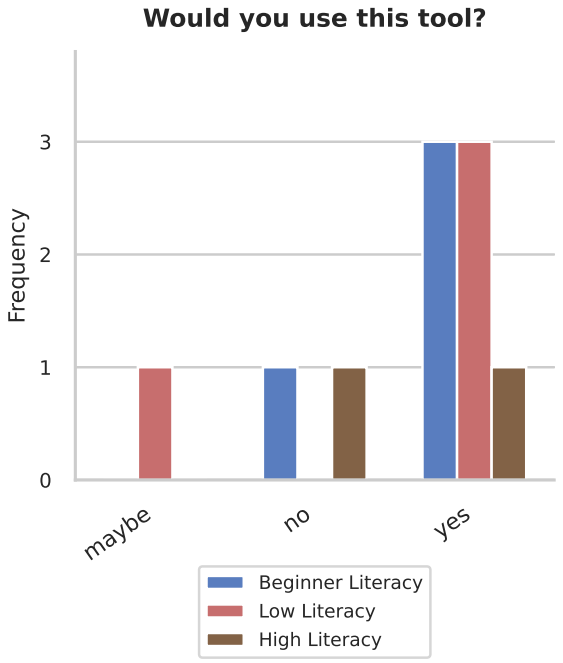

Figure 4: Graph of participants' responses to if they would use a tool like the prototype. Out of 10 participants, 7 responded 'yes', 1 responded 'maybe', and 2 responded 'no'.

helpful and wished the tool existed when they were attending Tamil language class during their adolescence. D, who self-described as conversationally fluent, thought that the pedagogical breakdown was not only helpful, but also corrects errors made by Google Translate. Lastly, R, who had only recently started learning Tamil, thought that the definitions provided by the tool were clearer than Google Translate.

4 User Requirements

Based on the results of the exploratory interviews, we established the following user requirements to guide the design of the website.

1. Intuitively understand the morphological breakdown even as an L2 learner.
2. Not overwhelm a beginner learner.
3. Is accessible on the phone.

We also drew inspiration from the Kawennón:nis verb conjugator for Kanyen'kéha (Kazantseva et al., 2018), (which itself was part of the inspiration for Gramble, discussed in Section 5.2), and its aim to assist users in acquiring morphology patterns rather than rote memorization of word definitions.

5 Systems Design

In this section we discuss the different components of Kelvi and how they interact, as visualized in Figure 5.

5.1 Datasets

We used two open source datasets for the Tamil dictionary component, specifically Tamil Wiktionary data Ylonen (2022) and dictionaries provided by Vasuki (n.d.).

5.2 Suffix Database Generation

The suffix database queried by the system was created using Gramble[2], a tabular programming language designed for linguistic parsing and generation. We used Gramble to aid us in generating a JSON file[3] consisting of Tamil suffixes, their meanings, and other useful (meta)linguistic information that are all stored in Gramble "tiers". In this section we limit discussion to what tiers (that are then converted into JSON attributes) we created for suffix generation and why.

The primary attributes that every suffix entry in the suffix database have are *text, gloss, display,* and *add-back*. *text* is a string representing a suf-

[2] https://nrc-cnrc.github.io/gramble/
[3] File can be found here.

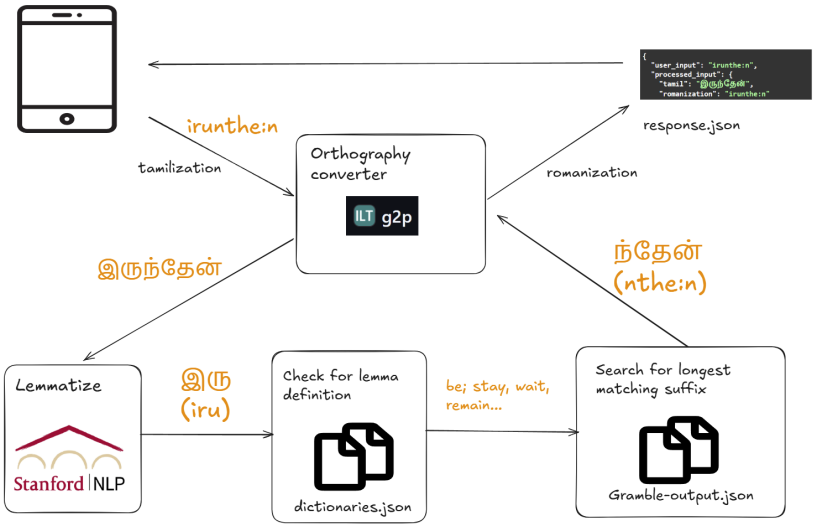

Figure 5: Diagram of Kelvi's backend and frontend.

Plural =		text	gloss	display	add-back
		கள்	[plural]	[கள்]	null
		ங்கள்	[plural]	[கள்]	ம்
		க்கள்	[plural]	[கள்]	null

Figure 6: Excerpt from Gramble file.

fix as it appears in an input word, and is the attribute that the input is matched against. For instance, the input பூதங்கள் *boothangaL* 'ghosts' would match to the suffix ங்கள் *ngaL*, as shown in Figure 6. *gloss* is a layman's definition of this suffix.

display is the version of the matched suffix that is shown to the user: this is generally the same as what is in *text*, but differs in situations where a suffix undergoes minor changes based on phonological environment[4]. For example, the plural suffix கள் *gaL* can also also appear as க்கள் *kkaL* or ங்கள் *ngaL* based on the sound it is preceded by, but since கள் is its prototypical form, this is the form of the suffix that is shown ("displayed") to the user in all cases, as shown in Figure 7. Finally, *add-back* is used as a secondary lemmatizer when the primary fails, to be detailed later in Section 5.5. In this instance, if the lemmatizer were unable to parse பூதங்கள், once the suffix match ங்கள் was identified and removed from the input, the character ம் (m) would then be "added back" to this new form to create the valid lemma பூதம் *bootham* ('ghost').

[4]Using linguistic terminology, *text* and *display* can roughly be equated to *surface form* and *underlying representation* respectively.

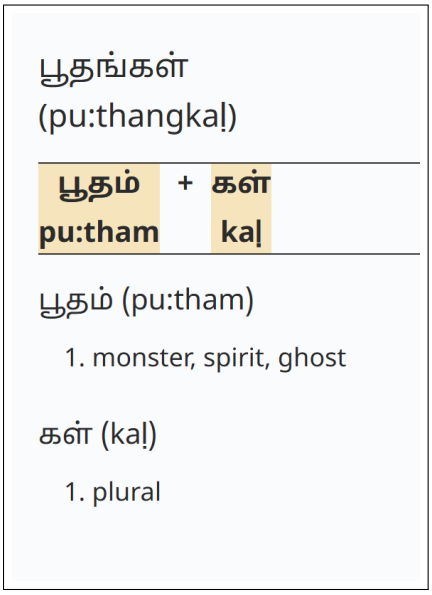

Figure 7: Kelvi UI when given பூதங்கள் *boothangaL* as input.

5.3 Romanized/Tamil Input

Users can input using the Tamil script or romanization (i.e., கேள்வி or *kelvi*). All of the dictionary and morphological data is stored in the Tamil script, so if a user inputs using romanization, their input is first converted into the Tamil script using a custom romanization-to-Tamil mapping that we wrote using G2P, an open-source grapheme-to-

phoneme transducer[5], (Pine et al., 2022) before being processed. Since there is no universal romanization standard followed by Tamil speakers, our mapping simply attempts to capture the most commonly used spelling patterns. After processing, the output is then returned to the user in both the Tamil script and romanization. The romanization presented here is derived from a Tamil-to-romanization mapping, again written using G2P. The romanization system used here is a combination of commonly used spelling conventions and IPA.

5.4 Lemmatization

Where possible, lemmatization of the Tamil input is performed using Stanza (Qi et al., 2020). The output lemma is then checked against the dictionary data to see if it has an existing dictionary entry. If no valid dictionary entry is found, other strategies are employed to identify the lemma, such as suffix matching.

5.5 Suffix Matching

To identify the potential suffix(es) on an input word, a backwards search is performed on the input to see if there is a match with any suffix included in the Gramble JSON file (generation of this file was discussed in Section 5.2). If there are multiple matches, the longest one (whose length is less than the length of the input) is selected. If a valid lemma was found using the lemmatization process described previously, we return both the lemma and the suffix that we identified. If a valid lemma wasn't found, we create a new potential lemma by removing the text of the matched suffix from the input word. If the potential lemma is now a valid lemma, (i.e., exists as a valid entry in the dictionary data), we return this alongside the suffix as the output to the user. If the potential lemma is still not a valid lemma, we check if the suffix that we found in the Gramble JSON has any material in its *add-back* field: this field contains phonological information that needs to be added to a stem in order to turn it into a lemma. If the *add-back* field is non-empty, we add this material to the potential lemma, confirm if this is now a valid lemma, and return it. If this strategy also fails, we return a "word not found" error to the user.

[5]https://github.com/NRC-ILT/g2p

6 Results

6.1 Systems Analysis

Due to limited resources and time (discussed in Section 9), we have not yet been able to perform a formal statistical evaluation. The minimal test set that was used to evaluate performance so far is composed of "edge-case" words, or words we predicted may be tricky to parse based on how our model is constructed. For instance, during early development, we noticed that the Stanford NLP lemmatizer often fails on class III verbs (as defined in Arden, 1910), so we made sure to explicitly compensate for this during the suffix generation process using Gramble, and included several words from this category as part of our testing. Since we observed the remaining six verb classes (Arden, 1910) being accurately lemmatized, fewer of these are represented in our test set as they were assumed to be higher performing. Similarly, there is a higher proportion of words with complex morphophonology and suffixal combinations present in our test set than is necessarily representative of the language, as we predicted these words to be the most challenging to cover. Thus, many "simple" words, for example words of the form [lemma + tense + person-number-gender], are not included in the test set, as once we confirmed that a sample few were working, we assumed others of the type would also be relatively high performing. Of course, formalized random testing is required to verify this. A high level overview of the tool's performance on this test set is shown in Table 1.

Result	# of Tokens	Percentage
Correct	141	50%
Incorrect	46	16%
No result	93	33%
Total	**280**	

Table 1: Performance of Kelvi on test input words

We primarily tested for presence of lemmas in our dictionary data (shown in Table 2), and accurate lemmatization and suffix identification (shown in Table 3).

Result	# of Tokens	Percentage
Correct	15	75%
Incorrect	1	5%
No result	4	20%
Total	**20**	

Table 2: Kelvi performance on inputs with no affixes

	# of Tokens	Percentage
Correct	126	48%
Incorrect	45	17%
No result	89	34%
total	**260**	

Table 3: Kelvi performance on inputs with affixes

For inputs that produced a "word not found" error, we isolated which were failing because the lemma itself was not present in our dictionary data, and which were failing because our tool was not able to lemmatize the word, as shown in Table 4.

Issue	# of Tokens	Percentage
Lemma not in data	10	11%
Bad lemmatization	79	89%
Total	**89**	

Table 4: Cause for "word not found" errors on inputs with affixes

This delineation will help us isolate which types of structures need to be added to our suffix database. A major category of words that are currently missing is lexical compounds. These are words that have two lemmas, which our model does not currently account for. Some lexemes commonly used in compounds like போது *po:thu* ('when', colloquially) can be added to the database as suffixes, but Tamil also has a productive process of forming novel compounds[6], that will require a more nuanced approach.

For inputs that produced incorrect outputs, we differentiated between whether the error was in the lemma, the suffix, or both (Table 5). The results

Issue	# of Tokens	Percentage
Incorrect lemma	9	20%
Incorrect suffix	32	71%
Both incorrect	4	9%
total	**45**	

Table 5: Types of errors on inputs with affixes

from Tables 4 and 5 in particular highlight that more work can be done in improving the suffix database, both to increase the accuracy and quantity of suffixes identified, as well as to assist in the lemmatization process. Thus this is where the focus of our future work lies.

[6]For example, the poetic word கண்ணுறங்கா *kaNNuRanga*: '(as) (my) eyes sleep' is composed of the lemmas கண் *kaN* 'eye' and உறங்கு *uRangu* 'to sleep', neither of which can be classified as "commonly" used in compounds.

6.2 User Feedback

We conducted a workshop for the launch of the MVP (minmum viable product) for Kelvi to gather user feedback. We also elicited feedback through word-of-mouth, and online through Reddit forums such as r/Tamil and r/LearningTamil.

The feedback received was largely positive. Users requested voice input, romanization input, including example sentences in the output, and providing the definition of the input word in its entirety (not just of the individual morphemes). We were able to add the feature of romanization input following this feedback, and are evaluating the feasibility of incorporating the other requested features; for instance, incorporating TTS (Mahaganapathy and Sarveswaran, 2025).

As discussed further in Section 9, due to lack of time we were not able to conduct a systematic user-feedback study post creation of the MVP. This would certainly ideally be part of a future study. Informally, co-author Srikanth has been able to use the tool as part of her regular Tamil teaching to positive effect. The major drawbacks currently are the low language coverage of the tool, especially when the input is in romanization. However, in the cases where the target words are present in Kelvi, learners found the provided definitions to be useful in understanding both the meaning of a word in its entirety and of its parts.

7 Discussion

Our current bottlenecks and proposed improvements to the software can be summarized as follows:

1. We have limited root-word (i.e., dictionary) data, and results can be improved if we are able to acquire more data.

2. The lemmatizer that we use fails on some inputs; as discussed in (5.5), we are compensating for these gaps using Gramble, which we will continue to do.

3. Some morphemes are homophonous; for example, the suffix -um on a verb means "it will", but on a noun means "and". Currently we return both definitions, but if we are able to acquire reliable part-of-speech data, we can refine the output returned to the user.

4. Users are able to input using both the Tamil script and romanization, but the romanization

input is significantly underperforming. This is because there is no romanization standard consistently followed by Tamil speakers, and so the mapping we currently employ cannot account for all possible spellings of a given Tamil word. The romanization practices of Tamil speakers are too irregular to be adequately captured by any one-to-many mapping[7], no matter how nuanced, and so this feature can only be improved if we are able to train a model on Tamil romanization input and/or implement an approximate search algorithm (Pine et al., 2025b, Littell et al., 2017).

5. Since Kelvi was primarily designed to capture written forms, words in the spoken/informal register are underrepresented. As these words vary widely across a range of dialects, collaboration with speakers from across this span of variation would greatly enhance the coverage of our tool. Currently, Kelvi is only designed to be useful in decoding the meaning of Tamil words inputted in the written/formal register, not for defining words expressed in the spoken/informal register or for converting spoken/informal forms into their written/formal equivalents.

Future work in this area may benefit from the use of machine learning techniques to address some of these challenges. As one example, we can potentially address the variation in romanization input by incorporating transliteration models, such as the one developed by Madhani et al., 2023, to convert between romanization and the Tamil script. Due to time constraints however, we were unable to properly investigate this approach.

8 Conclusion

Although Tamil has a relatively large speaker population, it is still considered a low-resource language which adds a layer of difficulty when developing digital tools and resources for Tamil that make use of data driven innovations in computational linguistics (Tissera and Saadhiq, 2025).

The system described in this paper serves as a starting point for creating pedagogical dictionaries designed for learners of polysynthetic languages and the morphological challenges they commonly face (Kazantseva et al., 2018). Although there is room for improvement in terms of coverage and accuracy, we hope that Kelvi serves as an example of a grassroots initiative that, despite limitations in terms of language data and resources, is able to create a useful language learning tool by taking a user-centered approach and targeting common pain points experienced by learners. The code and data used for the project is open source and will hopefully benefit other individuals and organizations looking to create language learning resources.

9 Limitations

The project received funding ($10,000 CAD) and had a 1 year deadline as per the agreement for the funding. Unfortunately, the cost of many high quality Tamil dictionary datasets far exceeded the available funding and consequently could not be used. This also impacted the testing process, as all test data had to be created manually and thus was limited to a few hundred inputs. Additionally, this project was done independently and on a voluntary basis without the affiliation of any research institution, which limited the time and resources that could be dedicated to the project.

10 Ethics

We discuss here the ethical considerations of this project on the axes of access, equitable representation of language variation, and data privacy and consent. First, however, we present the positionality of the authors.

One co-author (Srikanth) is a heritage Tamil speaker, and through their experience as a learner and teacher of the language, experienced both the challenge of developing their literacy skills and the lack of adequate resources online to help with this. Another co-author (Yu), is a heritage speaker of Cantonese and Mandarin and therefore also familiar with the challenges of heritage language learning. Additionally, both Srikanth and Yu have experience contributing to community-led projects to build digital tools for Indigenous language learning. These professional and lived experiences are

[7] In fact, it is impossible. This is because the letter "n" can be mapped to ண, ந, or ன. ண is pronounced differently from ந and ன and so can be mapped to a distinct romanized character like "N", but ந and ன are pronounced the same. They can't usefully be mapped to distinct romanized characters because even though they appear in complementary phonological environments 90% of the time, the remaining 10% the distinction is lexically determined, i.e. unpredictable from just the pronunciation, and therefore not something a user who can't read the Tamil script would be able to know.

what motivated the authors to create an online resource that could support learners of minoritized and low-resource languages.

10.1 Access

Kelvi was intentionally designed to be open-source and free to use because the motivation behind it was to develop a framework for a language learning tool that could be used by other low resource, minoritized languages. The intention was not to contribute to the history of extraction, exploitation, and gatekeeping of language resources in the field of linguistics (Leonard, 2021) by using language data and putting it behind a pay-wall, especially given that co-author Srikanth is a member of the language community the tool was built for. Open access to language data was deemed appropriate in this case, as Tamil enjoys majority language status in numerous jurisdictions (Tamil Nadu, Sri Lanka, Singapore), and so it wouldn't be considered sensitive information. In future collaborations with other language communities, however, those communities will always have full rights to their data and the choice to keep it open or limited access to only community members, and this is supported by the tool's permissive open source licence.

10.2 Equitable Representation of Language Varieties

We are aware that digital representations of language, especially in educational contexts such as Kelvi, can influence the relative prestige of the varieties used or omitted. Kelvi was deliberately designed to only represent the formal written variety of Tamil. The formal register remains largely uniform across the social axes of religion and region, along which conversely, the spoken register varies greatly. In this way, we hope that Kelvi is accessible to all Tamil users. That being said, we acknowledge that because spoken varieties are not represented in Kelvi, this may contribute to the commonly held belief of language purity that privileges written Tamil over spoken, informal Tamil (Annamalai, 2011). Currently our team is lacking in the linguistic expertise required to adequately represent all of the spoken varieties fairly, and so we seek to form collaborations with speakers of different Tamil varieties to make Kelvi more inclusive in this way.

While formal written Tamil does not undergo significant variation, the same is not true of Tamil written using romanization. There is no "standard" for romanization, and so each individual uses their own conventions. We don't want to present any one romanization system as a "standard"; in the romanization shown to the user, we use a convention that is associated more with academia, using characters that aren't easily typed (i.e., ṭ), to signal to users that the tool isn't meant to influence how they already type. However, the options for romanization input are very limited, and because the romanization is built using a one-to-many mapping (5.3), this mapping does inadvertently end up being prescriptive. Hopefully, as we improve the romanization input, this unintentional side-effect will lessen. Finally, since Kelvi is still only in the MVP stage, it does not have 100% accuracy yet. There are disclaimers on the website informing users of this, so hopefully this lets learners and teachers alike anticipate that there may be the occasional error.

10.3 Data Privacy and Consent

User input is temporarily available for us to view through the web hosting service Kelvi is hosted on. The input is anonymous and can't be tracked to any individuals. We view this input data solely for the purpose of bug-fixing (i.e., keeping track of any failing inputs to add to our testing process).

User feedback from the initial round of user testing has been anonymized, and none of their language data was used in the building of Kelvi or accessible to anyone outside of the team. The interviews were recorded with their consent and they were informed that the recordings would only be consulted for the purposes of improving the tool, and would not be shared with anyone else.

Additional feedback solicited once the MVP was built was encouraged to be submitted through the anonymized feedback forms on the Kelvi website. Some individuals contacted the authors directly; regardless, all feedback is kept anonymized here and in any future publications.

11 Acknowledgements

This project was made possible with support from the Audrey Wearn Language Prize and the Future Design School. We would also like to thank Fizza Ahmed, Vasumathi Srikanth, Srikanth Mangalam, Aidan Pine, Patrick Littell, Sowmya Vajjala, Roland Kuhn, and the Indigenous Languages Technology project team at the National Research Council of Canada. Finally, thank you to the anonymous reviewers for their feedback.

References

E. Annamalai. 2011. *Social dimensions of modern Tamil*, 1st edition. Cre-A:, Chennai.

E. Annamalai. 2019. Modern Tamil. In Sanford B. Steever, editor, *The Dravidian languages*, 2 edition. Routledge. Num Pages: 27.

E. Annamalai and R. E. Asher. 2002. *Colloquial Tamil: The Complete Course for Beginners*. Taylor & Francis Group, London, UNITED KINGDOM.

Albert Henry Arden. 1910. *A progressive grammar of common Tamil*. Madras : Society for Promoting Christian Knowledge.

Nathan Thanyehténhas Brinklow, Patrick Littell, Delaney Lothian, Aidan Pine, and Heather Souter. 2020. Indigenous Language Technologies & Language Reclamation in Canada. In *International Conference Language Technologies for All (LT4All): Enabling Linguistic Diversity and Multilingualism Worldwide*. UNESCO.

Mary Bucholtz. 2021. Community-Centered Collaboration in Applied Linguistics. *Applied Linguistics*, 42(6):1153–1161.

Elango Cheran. 2001. Learn Tamil.

Anna Kazantseva, Aidan Pine, Owennatekha Brian Maracle, and Ronkwe'tiyohstha Joe Maracle. 2018. Kawennón:nis: the Wordmaker for Kanyen'kéha (Ohsweken Mohawk). In *COLING: The 27th International Conference on Computational Linguistics.*

Wesley Y. Leonard. 2021. Toward an Anti-Racist Linguistic Anthropology: An Indigenous Response to White Supremacy. *Journal of Linguistic Anthropology*, 31(2):218–237.

Patrick Littell, Aidan Pine, and Henry Davis. 2017. Waldayu and Waldayu Mobile: Modern digital dictionary interfaces for endangered languages. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 141–150, Honolulu. Association for Computational Linguistics.

Patrick Littell, Darlene Stewart, Fineen Davis, Aidan Pine, and Roland Kuhn. 2024. Gramble: A Tabular Programming Language for Collaborative Linguistic Modeling. In *LREC-COLING 2024*. ACL.

Yanfei Lu, Patrick Littell, and Keren Rice. 2024. Empowering Oneida language revitalization: Development of an Oneida verb conjugator. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5757–5767, Torino, Italia. ELRA and ICCL.

Megan Lukaniec and Martin Holmes. 2023. Modern Wendat Lexicography: Using XML to Reflect the Grammar and Lexicon of an Iroquoian Language. *Dictionaries: Journal of the Dictionary Society of North America*, 44(2):75–105.

Yash Madhani, Sushane Parthan, Priyanka Bedekar, Gokul Nc, Ruchi Khapra, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Khapra. 2023. Aksharantar: Open Indic-language transliteration datasets and models for the next billion users. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 40–57, Singapore. Association for Computational Linguistics.

Irbha Magotra, Jyoti Sharma, and Supran Kumar Sharma. 2016. Assessing personal disposition of individuals towards technology adoption. *Future Business Journal*, 2(1):81–101.

Ahrane Mahaganapathy and Kengatharaiyer Sarveswaran. 2025. A survey and evaluation of text-to-speech systems for the tamil language. *Natural Language Processing Journal*, 12:100171.

Silvina Montrul. 2016. Heritage languages and heritage speakers. In *The Acquisition of Heritage Languages*, pages 13–40. Cambridge University Press, Cambridge.

Aidan Pine, Erica Cooper, David Guzmán, Eric Joanis, Anna Kazantseva, Ross Krekoski, Roland Kuhn, Samuel Larkin, Patrick Littell, Delaney Lothian, Akwiratékha' Martin, Korin Richmond, Marc Tessier, Cassia Valentini-Botinhao, Dan Wells, and Junichi Yamagishi. 2025a. Speech generation for indigenous language education. volume 90, page 101723.

Aidan Pine, David Huggins-Daines, Carmen Leeming, Patrick Littell, Timothy Montler, Heather Souter, and Mark Turin. 2025b. Zero-shot query generation for approximate search algorithm evaluation. In *Proceedings of the Eight Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 65–73, Honolulu, Hawaii, USA. Association for Computational Linguistics.

Aidan Pine, Patrick Littell, Eric Joanis, David Huggins-Daines, Christopher Cox, Fineen Davis, Eddie Antonio Santos, Shankhalika Srikanth, Delasie Torkornoo, and Sabrina Yu. 2022. Gi2Pi: Rule-based, index-preserving grapheme-to-phoneme transformations. In *5th Workshop on the Use of Computational Methods in the Study of Endangered Languages*. ComputEl-5.

Maria Polinsky and Olga Kagan. 2007. Heritage Languages: In the 'Wild' and in the Classroom. *Language and Linguistics Compass*, 1(5):368–395.

Arun Prasad. 2025. Tamil for Beginners.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Damodharan Rajalingam. 2025. rdamodharan/tamil-stemmer. Original-date: 2010-12-27T17:37:49Z.

Vasu Renganathan. 2019. Heritage language environment (HLE) and its implications upon language curriculum: a case of Tamil heritage students and designing a differentiated curriculum. In *Langues de l'Inde en diasporas : maintiens et transmissions*, Paris.

Kengatharaiyer Sarveswaran, Gihan Dias, and Miriam Butt. 2021. ThamizhiMorph: A morphological parser for the Tamil language. *Machine Translation*, 35(1):37–70.

Harold F. Schiffman, Vasu Renganathan, Prathima Christdas, S. Palanisamy, Thanam Baba, K. Karunakaran, Ganesh Krishnamurti, C.M. Pradeep, and S. Radhakrishnan, editors. 2009. *An English dictionary of the Tamil verb*. South Asia Studies Dept. University of Pennsylvania, Philadelphia, Pa.

Muditha Tissera and Hassaan Saadhiq. 2025. Natural Language Processing Resources for Tamil Language: A Systematic Review. *Journal of Information and Communication Convergence Engineering*, 23(4):236–258.

Vishas Vasuki. n.d. Dharma Via Vishvas.

Anne Wallisch, Olga Sankowski, Dieter Krause, and Kristin Paetzold. 2019. Overcoming fuzzy design practice: revealing potentials of user-centered design research and methodological concepts related to user involvement. In *2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, pages 1–9.

World Population Review. 2026. Official use of Tamil by country 2026.

Olcay Taner Yıldız, Begüm Avar, and Gökhan Ercan. 2019. An open, extendible, and fast Turkish morphological analyzer. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1364–1372, Varna, Bulgaria. INCOMA Ltd.

Tatu Ylonen. 2022. Wiktextract: Wiktionary as Machine-Readable Structured Data. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1317–1325, Marseille, France. European Language Resources Association.

A Interview Questions

Sociolinguistic questions used for the exploratory interview:

- How old are you?
- How would you describe your fluency in Tamil? What situations do you tend to use Tamil in? Speaking? Reading? Writing?
- Where did you go to school?

The following tasks were given to the participants to complete using the prototype. Note that the Tamil words were presented in text; no romanization was given and facilitators did not read out the words.

1. Find the meaning of பெரியவர்களுக்கு using the prototype interface. Try entering the word using Tanglish[8] / English letters.
2. Find the meaning of அறிந்துகொண்டேன் but type the word in Tamil.
3. Find the meaning of பெருகிவிட்டால் but use your voice.
4. Find the meaning of சிதறிவிடும் using your preferred input method from the previous tasks.

[8] Colloquial term used to refer to the code-mixing of Tamil and English in casual speech.