TAMZHI: Shorthand Romanized Tamil to Tamil Reverse Transliteration Using Novel Hybrid Approach

H M Anuja Dilrukshi Herath, T G Deshan K Sumanathilaka

Abstract— **Transliteration from Tamil to the Roman script holds a crucial place in the realms of effective communication, educational accessibility, and the seamless integration of digital technology. However, this process encounters a significant challenge due to the disparity in the number of vowels between the Tamil script, which encompasses a rich set of 12 vowels, and the Roman script, which is limited to just 5. This incongruity poses a substantial impediment when attempting ad-hoc transliteration of Tamil into the Roman script, especially when vowels are omitted. This paper aims to make a significant academic contribution by conducting an extensive literature review of recent developments in Romanized Tamil to Tamil transliteration, with a particular focus on addressing the absence of vowels. The review involves a meticulous examination of a wide array of methodologies proposed in recent years, ranging from rule-based systems to context-based strategies and machine learning-based approaches. In response to the challenges inherent in Tamil to Roman transliteration, this research work introduces a novel and innovative solution. This solution incorporates a Reverse Transliteration module, which leverages N-gram analysis and a rule-based model. The utilization of a trained trie structure is a key component of this approach, enabling word suggestions that effectively resolve ambiguities during the transliteration process. Remarkably, the proposed solution outperforms existing character-level transliteration methods, achieving an impressive character-level accuracy rate of 0.93. The practical implications of this research are substantial, particularly concerning the fulfilment of the linguistic and transliteration needs of native Tamil speakers within the digital platform, where such accuracy is of utmost importance.**

Keywords—**Hybrid Recommendation, N-gram, Rule-based, Suggestion, Transliteration, Tamil**

I. INTRODUCTION

Transliteration is the process of representing words or letters from one language using the characters of a different language [1]. It facilitates pronunciation for non-native speakers by utilizing a familiar alphabet. For instance, the Tamil word "அம்மா" is transliterated as "Amma" in English (Latin).

Unlike translation, which conveys meaning, transliteration conveys pronunciation in a different language. Conversely, reverse transliteration involves converting a word or sentence from a different language back into the original language. For example, the Tamil word "எப்படியாக," expressed in Romanized English as "epadiyaaga," can be reverse transliterated to its original form "எப்படியாக".

This study focuses on developing a reverse transliteration schema tailored to accommodate various typing patterns, including truncated forms without vowels. Given that Tamil is classified as a low-resource language, gathering the necessary transliteration data for the study presented inherent challenges. Consequently, this research predominantly relies on publicly available datasets and social media comments to collect the requisite transliterations. The primary objective of this research is to create a reverse transliteration model for Tamil. This model's purpose is to accurately convert Romanized Tamil words into their original Tamil forms, accommodating diverse typing patterns.

In an increasingly multilingual and interconnected world, the ability to convert Romanized Tamil words swiftly and accurately into their original form, even in cases where vowels are omitted, is more than a linguistic pursuit. It's a practical necessity. Tamil speakers often find themselves in situations where reverse transliteration is indispensable. For instance, in social media, where abbreviations and shortcuts are common, the ability to convert Romanized Tamil back into standard Tamil ensures a deeper understanding of messages and facilitates more meaningful interactions.

Consider social media, where communication transcends linguistic borders. Efficient conversion of Romanized Tamil, often abbreviated for convenience, back into regular Tamil provides a deeper understanding of sentiments and messages, fostering more meaningful interactions. In education, this research empowers learners and educators alike by ensuring accurate pronunciation and comprehension of Tamil. Tourists and travellers also benefit as they navigate and appreciate the nuances of a rich and ancient language.

In essence, this research bridges a critical gap in reverse transliteration capabilities, particularly for a language like Tamil, which presents unique linguistic challenges in both its script and its modern communication contexts. By achieving these objectives, we contribute not only to linguistic preservation but also to more effective communication and understanding in a diverse, interconnected world.

Tamil, being a low-resource language [2][3], lacks robust reverse transliteration solutions. This research aims to address this gap by creating an efficient and adaptable reverse transliteration model that will enhance cross-linguistic communication and transcription accuracy for Tamil. By doing so, it empowers users to effortlessly convert Romanized Tamil words, even in non-standard or truncated forms, back into their authentic Tamil counterparts. This development not only serves linguistic and cultural preservation but also opens doors to improved communication in a multilingual world.

Correspondence: Anuja Dilrukshi Herath (E-mail: anuja.20191180@iit.ac.lk)
Received: 01-08-2024 Revised: 18-03-2024 Accepted: 20-03-2024

Anuja Dilrukshi Herath and Deshan Sumanathilaka are from School of Computing, Informatics Institute of Technology, Colombo, Sri Lanka. (anuja.20191180@iit.ac.lk, deshan.k@iit.ac.lk).

DOI: https://doi.org/10.4038/icter.v17i1.7271

The contributions of the study can be highlighted as follows.

- Introduction of a Novel Hybrid Approach: Developed a unique transliteration system, TAMழ், that combines N-gram analysis, rule-based models, and trie structures to address the challenges in reverse transliterating Romanized Tamil to its original script, especially in cases where vowels are omitted.
- High Accuracy Rates: Achieved an impressive character-level accuracy of 93% and a word-level accuracy of 70%, surpassing existing character-level transliteration methods. This high degree of accuracy is critical for effective communication and comprehension in digital platforms.
- Ad-hoc Transliteration Handling: Innovatively addresses ad-hoc transliterations where vowels are omitted, a common occurrence in informal digital communication. This capability is especially important for understanding and participating in social media discussions, educational content, and digital communications in Tamil.
- Dataset Creation and Utilization: Assembled and utilized a novel dataset of Romanized Tamil sentences and their matching Tamil counterparts from social media content, enhancing the system's ability to deal with real-world transliteration challenges.

The upcoming sections will cover related work, describe the chosen methodology, present results, offer conclusions, and propose future research directions. Their purpose is to deepen comprehension of the subject and meaningfully advance the field.

II. Existing Works

Machine translation technologies have been adopted by many nations for their respective languages. The communication gap between speakers of different languages has been extensively studied in the fields of translation and transliteration. In the part that follows, we will investigate machine transliteration [11] strategies that are both focused on Tamil language transliteration and Indic language transliteration strategies.

A. Linguistic-based approach

This approach, often employed in machine translation, is commonly utilized for transliteration purposes. This strategy involves creating a system based on clear principles that are unique to the source language and any associated languages. This method falls into three primary categories: interlingual machine translation, transfer-based, and direct machine transliteration. Commercial transliteration tools like Azhagi and UCSC Transliterator frequently use rule-based methods. In a recent study by Dhariya [5] an error-correcting module was combined with the rule-based [8] technique for translation from Hindi to English code-mixed [16] text. Like this, Vijaya's [17] Tanglish to Tamil transliterator for city and people names followed a rule-based methodology but was unable to handle ad-hoc transliterations. This method's primary drawback is how significantly the transliteration process depends on a predetermined set of rules.

B. Statistical Approach

Machine Translation is all over widely used approach in statistical techniques are employed to understand the source and the target language's grammatical elements. This method enhances the accuracy and performance by breaking down sentences into n-grams [22]. By incorporating trigram, bigram, and unigram models into the architecture, the efficiency and reliability of the output are improved. In this study, the first stage of Tamil transliteration was conducted by the author using an n-gram model. A recent study by Zang [18] also utilizes an n-gram-based module combined with a lexical approach for transliteration, achieving a word-level accuracy of 0.74 in code-mixed texts from English to Hindi.

C. Knowledge-Based Approach

Knowledge model is a particular machine translation strategy that focuses on fusing machine translation tools with ontologies and the semantic web. This strategy has been used in many areas, including Tamil transliteration. A knowledge-based method has been successfully used by Sakuntharaj, R. and Mahesan [14] in an English-to-Tamil machine transliteration system that detects and corrects the spelling in Tamil text. To improve the precision and calibre of the Tamil language transliteration process, this method makes use of ontological knowledge and semantic linkages.

D. Example-Based Approach

The example-based [9] machine translation method for transliterating Tamil consists of three essential parts. The initial step involves balancing and handling the extracted.

linguistic data from the source. Subsequently, the relevant and isolated data is identified for translation. In the target language, it then integrates the translated information. Its ability to be used with any set of source and target languages, including Tamil, is one of the benefits of using this example-based machine translation approach. This method's efficacy depends on the accessibility and calibre of the example set [12] used to train the language model, which can greatly enhance the results of transliteration.

E. Neural-Based Approach

The neural-based technique, which makes use of neural networks, is a relatively recent addition to the family of machine translation (MT) approaches. Both long and short word sequences can be handled by neural models with ease. The target sentence's transliteration is then produced by the decoder. For machine translation, well-known neural models include LSTM [6], Imran [7] conducted recent research utilizing advanced transformer models such as BERT and RoBERTa. Their study implemented a Seq2seq model, which employed a bi-directional decoder. Future research in Tamil transliteration will have intriguing new directions thanks to further developments in neural machine translation [12], which hold the potential for more accurate transliteration.

F. Hybrid Approach

A recent advancement in machine transliteration (MT) is the hybrid strategy. To produce more accurate findings, it entails integrating two or more currently used MT techniques. The hybrid model [10] seeks to improve the transliteration process by utilizing the advantages of many techniques. Transliteration is carried out using a hybrid methodology in Sakuntharaj, R. and Mahesan [14] which combines tree-based algorithms with

the n-gram approach a knowledge base model. In this study hybrid approach was proposed to enhance the accuracy of Tamil transliteration. N-gram and tri-gram model, a linguistic model for back transliteration, a knowledge base approach for recommendation generation, and all of these are combined in this model. This integration of multiple approaches aims to improve the overall transliteration accuracy for Tamil language.

G. A comparison between the suggested solution and current techniques

Each method for machine transliteration has advantages and disadvantages. The rule-based method uses syntactic and semantic analysis to increase translation efficiency. But managing a lot of regulations can be time-consuming and expensive. Translations are more trustworthy thanks to the statistical approach, which uses corpus-based lexical resources. These systems do not have much adaptability, though. Although neural networks are effective at translating, they take a lot of computing resources to train on huge corpora. For neural models, ambiguity handling [4] throughout the translation phase presents a unique problem.

To create a lightweight transliterator for Tamil, using a hybrid ensemble approach that considers the benefits and drawbacks of previous methods. The hybrid model combines a Trie structure with the strength of N-gram models, rule-based methodologies, and knowledge bases. The Statistical approach and linguistic based methodology are effective in successfully transliterating both official and informal Tamil written in Roman characters. The Trie supported knowledge base successfully manages the word ambiguity in the Tamil language during the transliterator process. The hybrid model provides an all-inclusive solution for Tamil transliteration when used in combination.

III. METHODOLOGY

The study found that conducting the translation using a single approach is ineffective since ad-hoc transliterations won't be caught effectively. The author has decided to use a hybrid technique that includes transliteration so that the procedure can produce reliable findings.

A. Data Collection Approach

A dataset of "Romanized Tamil sentences" and their matching Tamil sentences that were taken from social media content was obtained [15]. The foundation for building the necessary transliterations for the study will be this processed dataset. The Dakshina dataset [13] from Google, which contains transliterations, will be used to enhance the set of transliterations required for the unigram tagger. The collection will include a variety of Romanized Tamil words and their corresponding Tamil words that were taken from social media networks. The process of creating a dataset includes a variety of topics, including social backdrop, music, politics, news, religion, and technology. The obtained dataset was manually translated into matching Tamil sentences. The necessary transliterations were produced and used in conjunction with the n-gram model using this well-vetted dataset.

B. Data Analysis

Online questionnaires [19] were given throughout several communities to collect typing patterns. To find various typing patterns, the replies from 215 users were gathered and analysed. To draw out useful patterns from the gathered data, segmentation, and alignment algorithms [20] were used. To build the rules required for the data annotation process, these discovered typing patterns were further studied. To track any changes in typing habits over time, three sentences were chosen from the original survey dataset and presented to a particular group of users once more. To determine the target audience and their associated typing styles within each group, stratified sampling approaches were used. Fig 1 shows the segmentation and alignment procedures used in this work to find the patterns.

V	A	N	A	K	K	A	M
V	-	N	-	K	-	-	M

Fig. 1 Data Annotation

C. Data Annotation

The data annotation technique was created using patterns that were discovered in the data analysis chapter. This method was created to produce both the appropriate Tamil words as well as Romanized Tamil words with various typing patterns. Vowel placement, consonant-vowel usage, and the mapping of English consonants to various Tamil characters were among the factors considered.

The data annotation technique was developed based on patterns identified during the data analysis chapter. These patterns served as the foundation for creating rules to generate appropriate Tamil words and Romanized Tamil words, encompassing various typing patterns. We considered factors such as vowel placement, consonant-vowel usage, and the mapping of English consonants to different Tamil characters. These findings culminated in the formulation of a rule-based transliterator, which encompasses 82 rules for vowels and consonants, 12 rules for handling hal symbols, and 16 rules for special characters.

To illustrate, let's delve into the rules for vowels and consonants. These rules define how specific Tamil characters should be transliterated into Romanized representations using a 12-character pattern scheme. For instance, the Tamil character "க" can be transliterated as "ka," "ந" as "na," and so on. Here, the rules account for the mapping of each Tamil character to its corresponding Romanized equivalent.

Hal symbols, which influence the pronunciation of Tamil words, are subject to specific rules in the transliteration process. For example, the Tamil hal symbol "்"signifies a consonant should not be followed by a vowel in pronunciation. Therefore, a rule is established to manage this, ensuring that in the Romanized representation, a consonant is not followed by a vowel. Special characters in Tamil words, such as the visarga "ஃ" or "ο" denoting the number zero, are addressed by unique transliteration rules. For example, the visarga "ஃ" may be transliterated as "ha," and "ο" as "0." These rules dictate how these specific characters are represented in Romanized form, preserving their distinct characteristics.

If we encounter the Tamil word "அம்மா," the rules dictate that each character should be transliterated as follows:

- "அ" transliterates to "a".
- "ம" transliterates to "ma".
- "்" is handled as per the hal symbol rule.
- "ம" transliterates to "ma".
- "ா" transliterates to "a".

Therefore, the transliteration of "அம்மா" becomes "amma" in Romanized form.

For a word like "கணக்கில்," the rules are applied as follows:

- "க" transliterates to "ka".
- "ண" transliterates to "na".
- "க" transliterates to "ka".
- "்" is handled as per the hal symbol rule.
- "க" transliterates to "ka".
- "ி" transliterates to "i".
- "ல" transliterates to "la".

Therefore, the transliteration of " கணக்கில்.

The Ad-hoc Transliteration Generator is the core engine responsible for generating Romanized representations of Tamil words based on the patterns and rules identified during the data analysis and annotation phases. Its functionality is multi-faceted, encompassing rule-based transliteration, word mapping, and the creation of a comprehensive Ad-hoc Romanized Tamil Dictionary.

By implementing these rules, the Ad-hoc transliterator generates an Ad-hoc Romanized [21] Tamil Dictionary. This dictionary comprises a comprehensive collection of Romanized representations of Tamil words, aligned with the transliteration rules and patterns defined by the generator.

The generator plays a pivotal role in mapping Tamil words to their Romanized counterparts. By following the established rules, it ensures that each Tamil word is accurately converted into its Romanized form, maintaining fidelity to the original pronunciation. This mapping forms the basis for building a comprehensive Ad-hoc Romanized Tamil Dictionary.

The Ad-hoc Romanized Tamil Dictionary serves as a valuable training resource for the Trie structure, a commonly used data structure for efficient word searching and retrieval. The Trie structure is trained using the generated transliterations from the Ad-hoc generator, which equips it with the ability to provide precise suggestions and mappings between Romanized Tamil words and their corresponding Tamil words, thereby facilitating effective cross-linguistic communication.

For a more detailed understanding, process flow of data annotation shown in Fig 2, and Table I presents a sample of the dataset. Table I further outlines the established guidelines for the ad-hoc Transliteration generator. Furthermore, identified rules for the ad-hoc Transliteral generator have been discussed in Table II.

Fig. 2 Data Annotation Process Flow

TABLE I

DATA ANNOTATION ALGORITHM RESULTS

வாருங்கள்	Varungal, Varungl, Vrungl, Vrngal, Varngal, Vrungl, Vrungl
கணக்கில்	Kanakkil, Kanakkl, Kanakil, Kankl, Knakil, Knkkil, Knkil, Knakl, Knkkl, Knkl
நில்லுங்கள்	Nillungal, Nilungal, Nillngl, Nllungl, Nllngal, Nlngl, Nllngl
யாருக்கு	Yaarukku, Yarukku, Yrukku, Yrkku, Yarkku, Yarkk, Yruku, Yrkku, Yrkk, Yrk

TABLE II

DATA ANNOTATION RULES

Rule Set	Rule Description
12-Character patterns	அ - A ஆ - Aa, Ae இ - i, e ஏ - e, a வ - wa, va ஷ - sha, sa த - tha, ta ட- dha, da ச - cha, ca ஓ - oo, o க – ka, ha ந - na
6 vowel combination	a, e, i, o, ae, ea
9 special character rule	ஃ - a, ah ஸ்ரீ - sri ஜ - ja ஸ - sa ஷ - sha ஹ - ha க்ஷ - ksha ற - ra ழ - zha, la

D. *Statistical approach with Tri-gram Tagger*

A dataset of texts in both Tamil and Romanized Tamil was used to train the N-gram [22] model. The sentences in the study were compiled by the author, who manually typed a portion of the dataset. In addition to the author's own typing, the study also utilized data from the Dakshina datasets [13], as well as supplementary data obtained from YouTube [16] and other online resources [15]. In addition to the dataset, the training process also used the annotated data produced by the annotation technique.

329500 words and 10,000 phrases were used in total for training. The Unigram Tagger [22], Bigram Tagger [22], and Trigram Tagger [22] with a backoff mechanism from NLTK [23] were used to implement the N-gram model. This approach ensures that higher-order n-grams can be employed when lower-order n-grams are not available, enhancing the accuracy of the model.

E. *Rule-Based Transliteration*

The Rule-Based Transliteration module is a crucial component of the transliteration system. The transliteration process commences by examining tokens with the transliteration "NNN." Each Romanized Tamil word

associated with the token undergoes pre-processing based on the patterns found in the ad-hoc transliteration schema. This pre-processing step ensures that the Romanized Tamil word is appropriately prepared for the transliteration procedure. To create the equivalent Tamil word required for the transliteration procedure, the established rule basis is employed. These rules, derived from pattern identification in the data annotation phase, ensure accurate and consistent transliteration from Romanized Tamil to standard Tamil.

F. Trie model Training with best word suggestion

The Trie [24] structure plays a pivotal role in the accurate transliteration of Romanized Tamil into standard Tamil. Here, we elaborate on the training process and word suggestion mechanisms within the Trie model.

1) Trie Model Training

The Trie structure was trained using the Ad-hoc Romanized Tamil patterns that were created by the annotation algorithms. The Python programming language was used to implement the Trie structure, and a full binary tree was utilized. Based on the Trie model the trained module can generate intermediate suggestions for words that have been romanized in Tamil. These intermediate terms in Romanized Tamil are then sent to the Knowledge Base, which is made up of several dictionaries with transliterations from Romanized Tamil to Tamil for different writing styles. Figure 3 depicts the flow of the model training and output-generating processes.

2) Best Word Suggestion

The trained Trie model, as previously mentioned, generates a list of potential Romanized Tamil words. To determine the most suitable term for users, a systematic selection process is employed. This process involves a comprehensive comparison of the suggested terms with the knowledge base, which contains the corresponding Tamil representations. Several metrics and algorithms for word selection are thoroughly discussed and applied. These metrics consider factors like accuracy, relevance, and contextual appropriateness. Users are then presented with a carefully curated set of suggestions, ensuring effective handling of ambiguity. Fig 3 illustrates the complete data flow of the Romanized Tamil to Tamil Transliteration procedure, shedding light on the intricacies of the word selection process.

IV. RESULTS

The BLEU score [20] has been used to evaluate the N-gram model with the Rule-based model, and word level accuracy has been used to evaluate the overall system. For evaluation purposes, test data from both the Thirukkural data set with 200 data records and the Dakshina dataset [13] with 500 data records were employed. While Dakshina's and Thirukkural dataset is geared towards documented is focused.

The following equations have been used to evaluate the whole system's word level (WL) and character level (CL) indicating accuracy.

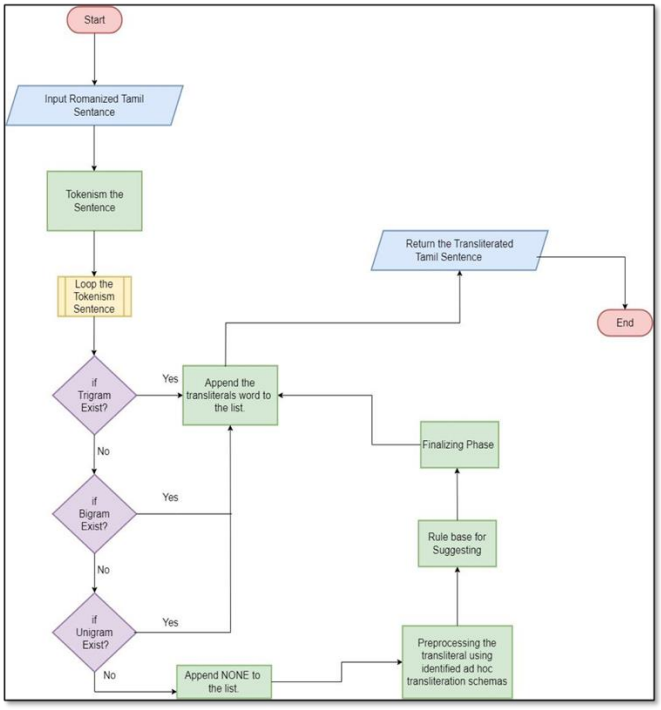

Fig. 3 Data flow Trigram, Rule base with suggestion level model

$$Accuracy\ (WL) = \frac{Correctly\ suggested\ words}{Total\ words} * 100\%$$

$$Accuracy\ (CL) = \frac{Correctly\ suggested\ character}{Total\ Characters} * 100\%$$

The system's accuracy was evaluated by employing test data from two prominent datasets, namely the Dakshina Dataset and Thirukkural dataset. Character-level accuracy and word-level accuracy were taken into consideration when rating the system's performance. In Table III below, the findings from these evaluations are summarised.

TABLE III

COMPARISON OF RULE BASED AND PROPOSED TRANSLITERATOR

Transliterator	Accuracy
Proposed Novel Transliterator TamZhi with Trigram + Rule based with Trie – Character level accuracy	93%
Proposed Novel Transliterator TamZhi with Trigram + Rule based with trie – Word level Accuracy	70%

The proposed TamZhi Transliterator demonstrated a remarkable character-level accuracy of 93%, highlighting its ability to provide precise character-level suggestions in the transliteration process. At the word level, it achieved an accuracy rate of 70%, indicating its proficiency in suggesting complete words accurately.

For comparison, the BLEU score, a metric widely used in machine translation and transliteration evaluations, was considered. The results of the proposed TamZhi Transliterator were compared to previous research efforts were shown in the below Table IV.

TABLE IV

COMPARISON OF PREVIOUS RESEARCH AND TAMZHI TRANSLITERATOR

Research	**BLEU Score**
Appicharla et al. (2021) (Microsoft) [4]	0.27
Raj and Laddagiri (2022) [12]	0.74
TamZhi Transliterator	0.86

As observed, the proposed TamZhi Transliterator achieves a BLEU score of 0.86. Unfortunately, the existing rule-based methods do not provide their accuracy metrics in terms of BLEU scores. However, these methods' performances are considered relatively low compared to the TamZhi Transliterator.

The character-level accuracy of 93% and the word-level accuracy of 70% for the TamZhi Transliterator can be viewed in the context of the absence of standards for what constitutes good or excellent accuracy for these metrics in the transliteration domain. Despite this limitation, a direct comparison with existing methods demonstrates that the TamZhi Transliterator significantly outperforms these traditional approaches, marking a substantial advancement in the field.

V. Discussion

In comparison to commercial products that employ a rule-based approach, the results generated by the proposed TamZhi transliterator were significantly improved. The accuracy of the proposed transliterator in terms of character-level accuracy reached 93%, while word-level accuracy reached 70%. This outperforms existing rule-based methods. Table V presents sample Romanized Tamil sentences and their transliterations by both the existing rule-based method and the TamZhi transliterator.

TABLE V

COMPARISON OF RULE BASED AND PROPOSED TRANSLITERATOR

Romanized Tamil Sentence	Existing Rule based method	TamZhi Transliterator
Puthiya vedhantham	புதிய வேதாந்தம்	புதிய வேதாந்தம்
Avi thenkoottai otha arugona vadivl araigal kondathaaga irukkum	அவை ஓத தேன்கூட்டை அறுவகாண வடிவில் அவறகள் ககாண்டதாக இருக்கும்	அவை தேன்கூட்டை ஒத்த அறுகோண வடிவில் அறைகள் கொண்டதாக இருக்கும்
Thorathil ulla oru grmamagm	வதாற்றத்தில் உள்ள ஒரு கர்மமாகும்	தூரத்தில் உள்ள ஒரு கிராமமாகும்

The proposed TamZhi transliterator demonstrates its superior performance compared to existing rule-based methods, achieving character-level accuracy of 93% and word-level accuracy of 70%. These results indicate the effectiveness of the approach in converting Romanized Tamil into its authentic form.

The system's high accuracy levels hold significant promise for applications in social media communication, education, and other domains, as it facilitates more accurate and meaningful interactions. By achieving these objectives, the proposed transliterator contributes to bridging a critical gap in reverse transliteration capabilities, especially for a language like Tamil, which presents unique linguistic challenges. This advancement not only serves linguistic and cultural preservation but also fosters improved communication in our diverse, interconnected world.

VI. Future work and Conclusion

The proposed Reverse transliterator performs better in the back transliteration process. The system obtained a word-level accuracy of 70% and a character-level accuracy of 93%. As a result, the TamZhi reverse transliterator can efficiently bridge the gap in Romanized Tamil to Tamil transliteration by employing ad-hoc transliteration rules. Users will be able to type Tamil using a vowel-free paradigm, a novel contribution to the discipline. This novel approach has applications in a variety of fields, including social media-based communication, education, and tourism-related applications. Notably, the TamZhi reverse transliterator offers a novel and practical feature, allowing users to type Tamil without vowels, a unique and valuable contribution to the field. This innovation opens doors to diverse applications, spanning social media-based communication, education, and tourism-related use cases. In the realm of social media, our system enhances cross-linguistic communication by making interactions more intuitive and meaningful. Users can now seamlessly transform Romanized Tamil texts, even in abbreviated forms, into authentic Tamil, ensuring a deeper understanding of messages and emotions conveyed in online conversations. In educational settings, our system empowers learners and educators by facilitating accurate pronunciation and comprehension of Tamil. It opens doors for Tamil speakers and learners to explore their language with confidence. Furthermore, travellers and tourists can benefit from our system as they navigate through the rich and ancient Tamil culture, appreciating its nuances without the constraints of complex transliteration. In future enhancements, several promising directions can further improve the system's capabilities. The system can be extended to accept code-mixed Romanized Tamil, which is currently a limitation. To address the disparity between character-level and word-level accuracy, a post-processing step may be introduced to refine the finalization phase by selecting the next word with the highest probability.

Moreover, automation of the data annotation algorithm through segmentation and alignment techniques can enhance the efficiency of dataset production. Furthermore, the adoption of a neural model based on a sequence-to-sequence (seq2seq) architecture holds the potential to elevate the reverse transliteration process to new levels of accuracy and performance.

As a next step, the system's effectiveness and generalizability can be assessed with other South Indian languages, leveraging the proposed architecture to explore broader applications in the region. By combining the strength of rule-based transliteration with the potential of advanced neural models and accommodating code-mixed languages, the TamZhi reverse transliterator paves the way for more accurate, adaptable, and versatile Romanized to Tamil transliteration systems, meeting the diverse needs of users in various domains.

Despite these promising directions, it's crucial to acknowledge the limitations. The TamZhi reverse transliterator is not without challenges, and there's an ongoing

quest to enhance its performance and adaptability to diverse linguistic contexts.

In conclusion, this work marks a significant advancement in Romanized to Tamil transliteration systems, addressing the varied needs of users in multiple domains. The journey towards greater accuracy, adaptability, and versatility in enhancing the Romanized Tamil to Tamil transliteration experience continues. While this study has achieved substantial progress, it also recognizes the challenges and opportunities that lie ahead, marking the beginning of an exciting and evolving field of research.

REFERENCES

[1] "Transliteration," Oxford American Dictionary. Oxford University Press,2010.

[2] Ramesh, A.; Parthasarathy, V.B.; Haque, R.; Way, A. Comparing Statistical and Neural Machine Translation Performance on Hindi-To-Tamil and English-To-Tamil. Digital 2021, 1, 86-102. https://doi.org/10.3390/digital1020007

[3] Amrhein, C. and Sennrich, R. (2020). On Romanization for Model Transfer Between Scripts in Neural Machine Translation. Findings of the Association for Computational Linguistics: EMNLP 2020. November 2020. Online: Association for Computational Linguistics, 2461–2469. Available from https://doi.org/10.18653/v1/2020.findings-emnlp.223

[4] Appicharla, R. et al. (2021). IITP-MT at WAT2021: Indic-English Multilingual Neural Machine Translation using Romanized Vocabulary. Proceedings of the 8th Workshop on Asian Translation (WAT2021). August 2021. Online: Association for Computational Linguistics, 238–243. Available from https://doi.org/10.18653/v1/2021.wat-1.29

[5] Dhariya, O., Malviya, S. and Tiwary, U.S. (2017). A hybrid approach for Hindi-English machine translation. 2017 International Conference on Information Networking (ICOIN). 2017. Da Nang, Vietnam: IEEE, 389–394. Available from https://doi.org/10.1109/ICOIN.2017.7899465

[6] Dhivyaa, C.R. et al. (2022). Transliteration based Generative Pre-trained Transformer 2 Model for Tamil Text Summarization. 2022 International Conference on Computer Communication and Informatics (ICCCI). January 2022. 1–6. Available from https://doi.org/10.1109/ICCCI54379.2022.9740991

[7] Imran. (2018). So what's the catch with LSTM? 2 February. Available from https://datascience.stackexchange.com/questions/27392/so-whats-the-catch-with-lstm.

[8] Kavirajan, B. et al. (2017). Improving the rule-based machine translation system using sentence simplification (english to tamil). 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). September 2017. 957–963. Available from https://doi.org/10.1109/ICACCI.2017.8125965

[9] M. Mayavathi, K. Deepa, and Bharath Niketan. (2012). QUALITY TRANSLATION USING THE VAUQUOIS TRIANGLE FOR ENGLISH TO TAMIL. Available from http://uttamam.org/papers/12_39.pdf

[10] Mathur, S. and Saxena, V.P. (2014). Hybrid appraoch to English-Hindi name entity transliteration. 2014 IEEE Students' Conference on Electrical, Electronics and Computer Science. March 2014. 1–5. Available from https://doi.org/10.1109/SCEECS.2014.6804467

[11] Oh, J.-H., Choi, K.-S. and Isahara, H. (2006). Improving Machine Transliteration Performance by Using Multiple Transliteration Models. In: Matsumoto, Y. Sproat, R.W. Wong, K.-F. et al. (eds.). Computer Processing of Oriental Languages. Beyond the Orient: The Research Challenges Ahead. Lecture Notes in Computer Science. 2006. Berlin, Heidelberg: Springer, 85–96. Available from https://doi.org/10.1007/11940098_9

[12] Raj, Y. and Laddagiri, B. (2022). MATra: A Multilingual Attentive Transliteration System for Indian Scripts. Available from http://arxiv.org/abs/2208.10801

[13] Roark, B. et al. (2020). Processing South Asian Languages Written in the Latin Script: the Dakshina Dataset. Available from http://arxiv.org/abs/2007.01176

[14] Sakuntharaj, R. and Mahesan, S. (2016). A novel hybrid approach to detect and correct spelling in Tamil text. 2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS). December 2016. 1–6. Available from https://doi.org/10.1109/ICIAFS.2016.7946522

[15] Transliteration of Secured SMS to Indian Regional Language | Elsevier Enhanced Reader. (no date). Available from https://doi.org/10.1016/j.procs.2016.02.048

[16] Vasantharajan, C. and Thayasivam, U. (2022). Towards Offensive Language Identification for Tamil Code-Mixed YouTube Comments and Posts. SN Computer Science, 3 (1), 94. Available from https://doi.org/10.1007/s42979-021-00977-y

[17] Vijaya, M. et al. (2009). English to Tamil Transliteration using WEKA | Deva kumar - Academia.edu. Available from https://www.academia.edu/51050763/English_to_Tamil_Transliteration_using_WEKA?from_sitemaps=true&version=2

[18] Zhang, Y. et al. (2018b). [PDF] A Fast, Compact, Accurate Model for Language Identification of Codemixed Text | Semantic Scholar. Available from https://doi.org/10.18653/v1/D18-1030

[19] H. Anuja, "A Survey to Collect Romanized Tamil Typing Patterns." https://docs.google.com/forms/d/1xycdES_ZwcJTxyPvfqoZLjlOWnTSUeerlsu1TiBBXE8/edit

[20] B. Roark et al., "Processing South Asian Languages Written in the Latin Script: the Dakshina Dataset," p. 11, 2020.

[21] T. G. D. K. Sumanathilaka, R. Weerasinghe and Y. H. P. P. Priyadarshana, "Swa-Bhasha: Romanized Sinhala to Sinhala Reverse Transliteration using a Hybrid Approach," from http://doi:10.1109/ICARC57651.2023.10145648.

[22] Manning, Christopher & Utze, Hinrich & Lee, Lillian. (2000). Foundations of Statistical Natural Language Processing, from https://nlp.stanford.edu/fsnlp/

[23] Bird, Steven & Klein, Ewan & Loper, Edward. (2009). Natural Language Processing with Python, https://www.nltk.org/book/

[24] Ma, D., Feng, J. (2014). A Generic Approach for Bulk Loading Trie-Based Index Structures on External Storage. In: Li, F., Li, G., Hwang, Sw., Yao, B., Zhang, Z. (eds) Web-Age Information Management. WAIM 2014. Lecture Notes in Computer Science, vol 8485. Springer, Cham. https://doi.org/10.1007/978-3-319-08010-9_8