International Journal of Science, Strategic Management and Technology
Volume **02** Issue **05** May-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

Smart Tamil: A Dialect-Aware Small Language Model for Tamil NLP

Gokul K
Kishore Kumar R
Tholkappiyan R
UG Student,
Department of CS & IT,
Vels Institute of Science, Technology And Advanced Studies
(VISTAS), Pallavaram, Chennai-600117, Tamil Nadu, India

Dr. R PADMA
M.Sc.,M.Phil.,PGDHRM.,SET.,Ph.D.,
Assistant Professor, Department of CS & IT,
Vels Institute of Science, Technology And Advanced Studies
(VISTAS), Pallavaram, Chennai-600117, Tamil Nadu, India

https://doi.org/10.55041/ijsmt.v2i5.046

Cite this Article: K, G., R, K. K. & R, T. (2026). Smart Tamil: A Dialect-Aware Small Language Model for Tamil NLP. International Journal of Science, Strategic Management and Technology, *02*(05). https://doi.org/10.55041/ijsmt.v2i5.046

ABSTRACT

Smart Tamil, designed to be a dialect-sensitive language system, aims to embrace the diversity and richness of Tamil as spoken in the dialects of Tamil Nadu. Most language systems do not take dialect differences explicitly during large-scale deployments and the output language is grammatically correct but evocative of a non- idiomatic language. To solve this, the Small Language Model (SLM) of Smart Tamil will be trained on small- corpus spoken data, heterogeneous written data, and video data to capture the language and dialect variations and spoken styles of the five major dialect zones of Tamil Nadu including Kongu Tamil (Coimbatore/Erode), Nellai Tamil (Tirunelveli/Thoothukudi), Kanyakumari Tamil, the Central Trichy/Thanjavur, and Urban Tamil of Chennai. The Smart Tamil System has been built as a full stack React + Flask application with the inbuilt ability for speech synthesis, and speech recognition through the Web Speech API.

Index Terms *— Tamil NLP, dialect-aware language model, small language model, speech recognition, regional language processing.*

1. INTRODUCTION

Tamil is one of the oldest classical languages in the world and the language of more than 75 million people in Tamil Nadu and the numerous communities in the Srilankan state of the North, Singapore, Malaysia and across the world. Despite Tamil being spoken in diverse dialects, the language and dialect variations of spoken Tamil and Tamil vernaculars, and the social/cultural dimension, remains underexplored in the NLP community. Chennai Tamil is characterized by being fast and urban, with code switching to English; Madurai Tamil is slow and patient and characterized by adherence to traditional respect forms with social warmth; the Kongu belt (Coimbatore, Erode) has frequent spoken contractions; Tirunelveli Tamil is rhythmic (percussive) and expressive; Kanyakumari Tamil is epistemic; and Trichy/Thanjavur Tamil are more endemically located within the language — these dialects are the most equidistantly neutral, standard dialects of Tamil.

NLP systems often conceptualize Tamil as one static language with uniform output. Tamil is incredibly diverse, and within the fields of Tamil speech and lexicon, Tamil is still deeply characterized grammatically accurately but without the spoken idioms often interchangeable. The output is often grammatically correct, but the systems have insensitivity

towards regional pulse and fail to accommodate the idioms specific to the Tamil dialects of Tamil Nadu. There is sometimes a bias towards one dialect over others.

2. LITERATURE SURVEY

2.1 NLP Development

NLP went through several cycles in the 1980s/90s, starting with rule-based systems, transitioning to model-based methods, then statistical methods. BERT established state-of-the-art performance for bidirectional language understanding; GPT demonstrated autoregressive generation at scale. The caveat: almost all development was focused on English and a few high-resource languages. Even in Tamil NLP — aside from some strides forward in machine translation, sentiment analysis, and speech recognition — it has become tethered to formal written corpora, completely ignoring dialectal and spoken varieties.

2.2 Small Language Models

Large language models have a very strong standardisation prior, which faces resistance from dialectal fine-tuning. Unlike SLMs trained on specifically curated dialect data, larger models fail to genuinely adapt. SLMs trained correctly on dialectal data represent those dialects as authentically as the data reflects the dialect itself, and remain deployable on ordinary hardware without specialised infrastructure.

2.3 Dialectal Variations and Their Linguistic Dimensions

Dialect variations in the Tamil language function at several levels. There is phonological variation where pronunciations vary with regions based on differences in vowel lengths, consonant pronunciations, and rhythm. Lexical variation involves distinct words used to describe one entity or object across dialects. Differences in suffixes — such as Kongu "-nga" indicating warmth — constitute morphological variations. There are syntactical variations in the way questions and clause structures are formed. All of these variations have remained unseen by conventional NLP systems trained on normalized texts.

2.4 Voice Technology and the Dialect Variation Problem

Most current ASR technology has been developed around urban and educated speakers using standardised pronunciation. A Tirunelveli speaker in a noisy environment receives consistent misrecognition while a Chennai speaker is recognized correctly. Similarly, TTS produces prosody standardised to standard Tamil, making it sound alien to regional speakers. Building a dialect-sensitive language model will greatly aid in improving speech recognition and synthesization capabilities.

3. OBJECTIVES AND METHODOLOGY

3.1 Core Objectives

- Develop a dialect-aware SLM that learns the phonology, lexicon, morphology, and style of five distinct Tamil dialects rather than just replacing surface vocabulary.
- Facilitate paraphrasing into dialects: take any input sentence and generate equivalent outputs in every target dialect while ensuring meaning equivalence using automatic semantic similarity measurement.
- Provide a comprehensive, speech-enabled full-stack application framework (React front-end, Flask back- end, SQLite3 database, Web Speech API) compatible with generic computer hardware without specialised infrastructure.

3.2 Methodology Overview

The method is executed in ten steps: (1) Data Gathering from local media, conversation records, and community-produced digital text, with systematic geographically distributed sampling in all five dialect areas. (2) Preprocessing and Annotation: transcribe audio files by dialect-aware human annotators, cleanse texts, perform intra-dialect normalisation while maintaining dialectal features, and annotate their regional origin, dialect identifier, and linguistic attributes. (3) Model Choice and Fine-tuning: select an SLM pre-trained on Tamil language foundations, then fine-tune iteratively on dialectally annotated data with learning rate decay and early stopping strategy. (4) Dialect Conditioning: train region-specific conditioning signals to associate each signal with coherent dialect-specific outputs.

4. SYSTEM DESIGN

4.1 System Architecture

Smart Tamil is a full-stack dialect-aware NLP system. The React front-end manages text and voice input, dialect selection marked with identifiable location names, and output displays side-by-side across selected regions. The Flask back-end processes requests by verifying them, executing inference on the selected dialects, and generating paraphrases. SQLite3 provides storage for dialect, session, and output data.

4.2 Data Acquisition and Preprocessing

The data used for analysis comes from three different sources: local media (YouTube channels in the region, community radios, regional news broadcasts), recordings of conversations acquired with proper permission, and digital content created by the community (social media, discussion forums, instant messages). The geographic coverage includes Coimbatore/Erode (Kongu), Tirunelveli/Thoothukudi (Nellai), Kanyakumari, Trichy/Thanjavur (Central), and Chennai.

4.3 Dialect-Specific Paraphrasing

This module is fed a sentence along with instructions to maintain semantic integrity and information specifying which dialect should be used as context. Output sentences are automatically scored for semantic similarity, with sentences that diverge from the original meaning filtered out. Human evaluation of selected output sentences detects pragmatic deviations not caught by the automated system. Six or seven variants of a sentence can be generated in one API call.

5. RESULTS AND DISCUSSION

5.1 Quantitative Results

The fine-tuned dialect-aware SLM showed clear and consistent improvement over the base model across all evaluated regions on both BLEU and ROUGE metrics. Dialects with the richest training data (Chennai, Madurai, Tirunelveli) showed the strongest performance. Importantly, semantic similarity scores for paraphrased outputs remained consistently high even where BLEU scores were more modest — indicating that the model learned semantic preservation more robustly than stylistic adaptation in some cases, which is the preferable failure mode.

5.2 Qualitative Analysis — Native Speaker Evaluation

Native Tamil speakers from each target region assessed outputs on fluency, authenticity, and contextual accuracy. Fluency ratings were significantly higher for the fine-tuned model across all regions — evaluators consistently described outputs as natural and conversational rather than formal and institutional. Authenticity ratings were strong for well-represented dialects: Tirunelveli outputs showed correct rhythmic energy and vocabulary; Kongu outputs used the "-nga" suffix naturally; Kanyakumari outputs showed recognisable Malayalam-influenced features. For less-represented dialects, evaluators recognised the dialectal intent but noted that finer rhythmic and hyper-local details were sometimes slightly off. Semantic accuracy of paraphrases was rated high across the large majority of outputs.

5.3 Comparative Analysis

Dimension	Standard Tamil Model	Dialect-Aware SLM
Dialect Awareness	Low — no regional differentiation	High — distinct output per region
Output Naturalness	Moderate — formal register	High — conversational register
Paraphrasing Quality	Low — meaning drift frequent	High — meaning preserved
Native Speaker Acceptance	Low — described as formal/stiff	High — recognised as authentic
Computational Efficiency	Moderate	High — optimised for inference

Table I: Comparative Analysis — Standard Tamil Model vs. Dialect-Aware SLM

5.4 Application Screenshots

The following screenshots demonstrate the Tamil Words Explorer application running at localhost:3000.

Fig. 5.3: Tamil Words Explorer — Hero Section with Bilingual Voice Search Interface

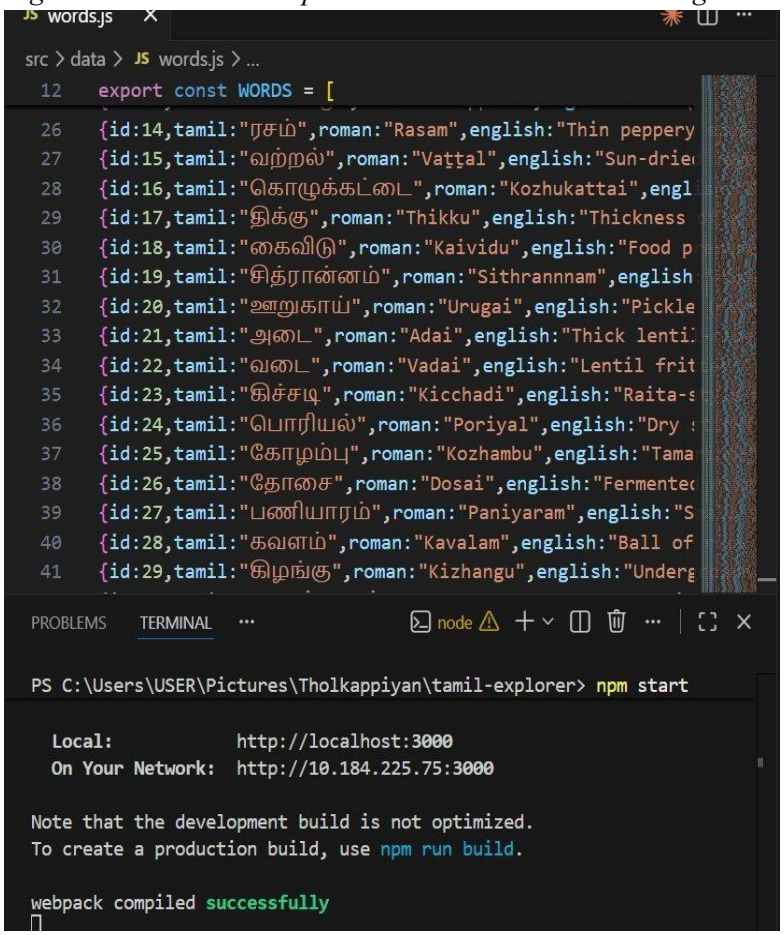

Fig. 5.4: VS Code — words.js Data File with 29+ Tamil Food Words Dataset

Fig. 5.5: Word Cards Grid — Category Filter Bar with 600 Tamil Words Across 8 Categories

*Fig. 5.6: Voice Recognition Active — Tamil Speech Input (*கேட்கிகேன்*...)*

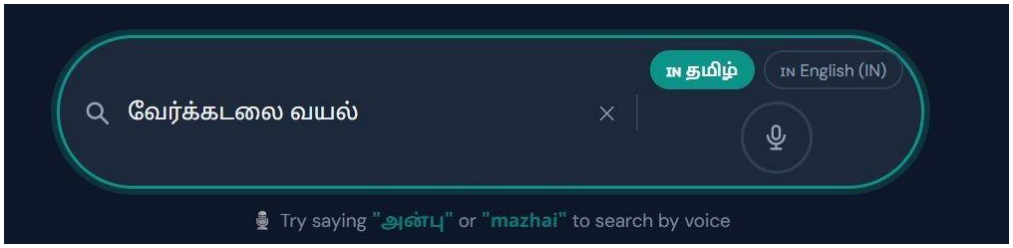

Fig. 5.7: Voice Search Result — கேர்ேஂேடலை ேயை *Typed in Tamil Mode*

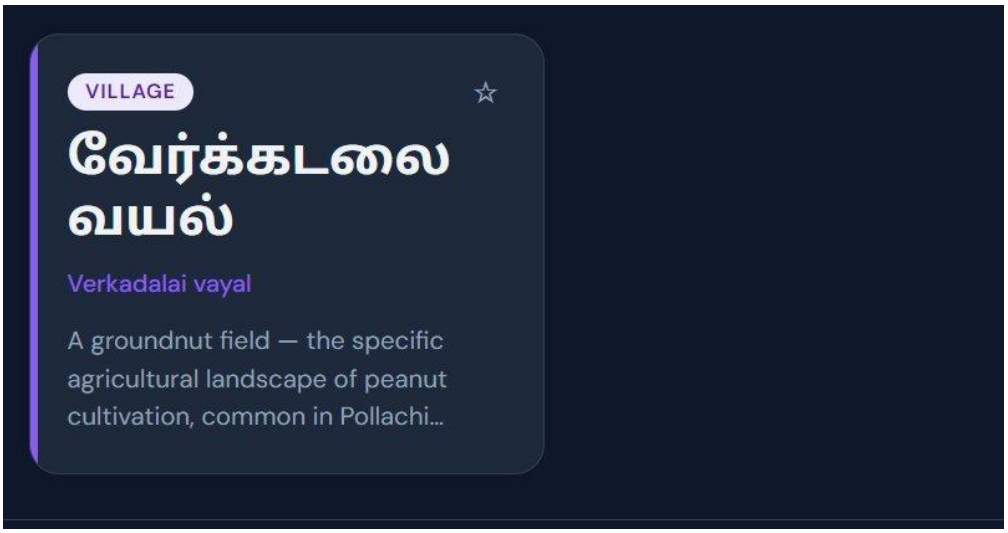

Fig. 5.8: Word Detail Card — கேர்ேஂேடலை ேயை *(Verkadalai Vayal) Village Category*

5.5 Key Findings and Limitations

The most important finding: data quality and coverage determine performance more than any architectural choice. Dialects with richer training data performed best on every measure. The SLM approach validated itself — adapting more genuinely to dialectal data than a larger, more opinionated model would, while remaining deployable on ordinary hardware. Human evaluation is irreplaceable; BLEU and ROUGE undervalue authentic but lexically diverse outputs and miss rhythmic and register dimensions that native speakers respond to most strongly.

Key limitations include: performance gap between text and speech inputs (browser-based ASR struggles with regional accents); uneven dialect coverage creates an equity gap; code-mixed Tamil (the natural speech mode of many urban speakers) is handled less reliably than pure Tamil; and performance degrades on long or syntactically complex sentences.

6. CONCLUSION AND FUTURE WORK

6.1 Conclusion

Smart Tamil demonstrates that dialect-aware Tamil language technology is technically feasible and practically valuable. The fine-tuned SLM produces outputs that are meaningfully more natural, more regionally authentic, and more recognisable to native speakers than a standard Tamil model. The dialect-based paraphrasing capability — producing six or seven regionally adapted semantic equivalents from a single input — is a technically significant achievement with real-world applicability in health communication, civic information, and education.

The choice of a Small Language Model was validated in practice: it adapts more genuinely to dialectal data, runs efficiently on ordinary hardware, and the full-stack application makes its capabilities accessible to real users. Most importantly, the results confirm the underlying premise: Tamil's dialectal diversity is worth taking seriously, and users notice and respond to the difference between generic Tamil and Tamil that sounds like theirs.

6.2 Future Work

Priority directions for future work: (1) Expand dialect coverage to underrepresented districts through fieldwork with local schools, cultural organisations, and community radio stations. (2) Build dialect-specific ASR and TTS systems trained on the same dialectal audio, so that voice interaction reflects regional prosody as well as vocabulary. (3) Address code-mixed Tamil with training data that represents the full range of Tamil-English mixing patterns. (4) Implement retrieval-augmented generation for hyper-local expressions. (5) Integrate continuous learning so the model can incorporate new patterns without full retraining cycles. (6) Extend the methodology to other Indian languages — the data collection, annotation, SLM fine-tuning, and evaluation approaches are applicable to any language with internal dialectal diversity.

REFERENCES

[1] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems, vol. 30, pp. 5998– 6008, 2017.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, pp. 4171–4186, 2019.

[3] T. Brown et al., "Language models are few-shot learners," in Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901, 2020.

[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] P. Joshi et al., "The state and fate of linguistic diversity and inclusion in the NLP world," in Proc. ACL 2020, pp. 6282– 6293.

[6] G. Murugesan et al., "Tamil NLP: Challenges, datasets, and deep learning approaches," Journal of Intelligent Systems, vol. 29, no. 1, pp. 498–509, 2020.

[7] R. Krishnamurthy, "A study of dialects in Tamil Nadu: Sociolinguistic perspectives," Indian Linguistics, vol. 70, no. 1–4,
pp. 1–25, 2009.

[8] A. Raj and S. Thomas, "Low-resource dialect adaptation using transfer learning for Dravidian languages," in Proc. ACL 2021.

[9] S. Soundararajan and B. Raju, "Regional dialect identification in Tamil using phonological features," ACM Trans. Asian Low-Resource Language Inf. Process., vol. 21, no. 4, pp. 1–20, 2022.

[10] D. Ghosh and R. Bhatt, "Code-mixing in South Asian languages," in Proc. ACL Workshop on Code-Switching, 2021.

[11] V. Gandhi and S. Dave, "Challenges in building NLP applications for Indian languages," in Proc. LREC-2020.

[12] E. Annamalai, "Nativization of English in India and its effect on multilingualism," Journal of Language and Politics, vol. 10, no. 1, pp. 71–87, 2011.

[13] R. Bommasani et al., "On the opportunities and risks of foundation models," arXiv:2108.07258, 2021.

[14] T. Mikolov et al., "Distributed representations of words and phrases and their compositionality," in Advances in Neural Information Processing Systems, vol. 26, pp. 3111–3119, 2013.

[15] M. Artetxe et al., "Efficient large scale language modeling with mixtures of experts," in Proc. EMNLP 2022.