TamilMayangoliSpell: An Open-Source Neural Framework for Context-Sensitive Mayangoli Error Correction in Tamil

Yazhmozhi VM[1] **Annalu Waller**[2] **Jacky Visser**[3]
[1]Computing, Edinburgh Napier University International College
[2]School of Science, Engineering and Environment, University of Salford
[3]Computing, University of Dundee
Correspondence: y.murugesan@napier.ac.uk

Abstract

Mayangoli errors are context-sensitive errors in Tamil that arise from confusion among phonetically similar graphemes (e.g., ல/ள/ழ, ர/ற, ந/ன/ண). These errors are challenging for conventional spell checkers because both incorrect and correct forms are valid dictionary words, making dictionary lookup insufficient and requiring contextual modelling. We present **TamilMayangoliSpell**, a reproducible framework for Mayangoli error correction that combines (i) Tamil-specific preprocessing for sentence segmentation and normalisation, (ii) linguistically grounded error induction for generating training data constrained by dictionary validity, and (iii) fine-tuning of multilingual sequence-to-sequence models. Using 30,000 sentence pairs derived from TamilCorp, a massive multi-genre Tamil corpus and split 80/10/10 into train/validation/test, we fine-tune mBART, mT5, and NLLB under a small hyperparameter grid using greedy decoding with a maximum sequence length of 128. mT5 achieves the best performance (BLEU 99.28; Exact Match Accuracy 93.50%) and remains strong in a cross-genre evaluation on short stories. The preprocessing scripts, generated parallel datasets, and trained models are publicly available in a GitHub repository.

1 Introduction

Context-sensitive real-word errors are valid dictionary words that become incorrect in a given context. Such errors are non-trivial to identify in morphologically rich and inflectional languages like Tamil without context modelling. This is concerning for *Mayangoli* errors: context-sensitive errors caused by confusion among phonetically similar graphemes (e.g., ல/ள/ழ, ர/ற, ந/ன/ண). A dictionary might contain குளம் (pond) as well as குலம் (clan). When a Tamil writer has used one for the other in an unrelated context, this is an instance of a Mayangoli error.

Existing works on the detection and correction of Mayangoli errors in Tamil have predominantly used dictionary-based or statistical methods (Segar and KengatharaiyerSarveswaran, 2015; Sithamparanathan and Uthayasanker, 2019), with limited reproducibility due to the lack of publicly available datasets and implementations (Sampath and Shanmugavel, 2023; Rajalakshmi et al., 2023). To address this gap, we propose **TamilMayangoliSpell**, an open-source neural framework for Mayangoli error correction that combines a linguistically grounded error-induction pipeline with multilingual sequence-to-sequence (Seq2Seq) models.

Contributions:

- A linguistically grounded error induction pipeline for generating Tamil context-sensitive Mayangoli errors using confusion-group substitutions constrained by dictionary validity.
- A controlled empirical comparison of multilingual Seq2Seq models (mBART, mT5, and NLLB) for Tamil orthographic error correction under identical training and decoding settings.
- Cross-genre generalisation, demonstrating that the learnt correction behaviour transfers beyond the training domain.
- Code, preprocessing scripts, trained models, and experiment configurations are published on GitHub[1] to support reproducible research in Tamil NLP.

2 Background

Classical Tamil grammar books like the *Tholkaappiyam* (Pillai, 1975) describe phonological distinctions, explaining confusions between graphemes.

[1]https://github.com/TamilGeekGirl/TamilMayangoliSpell

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 42–51

The major types of context sensitive errors in Tamil are summarised in this section, with a focus on Mayangoli errors, which form the scope of our work.

2.1 Context-sensitive errors

Context-sensitive errors are valid words that are irrelevant to the context in which they are used. They differ from non-word errors (words not in the lexicon/dictionary), which are often detectable via dictionary lookup. In Tamil, context-sensitive errors can go undetected because Tamil orthography supports many pairs that are both valid dictionary words and differ by a single character.

Context-sensitive errors in Tamil include *Kuril–Nedil* errors (short vs. long vowel confusions) and *Mayangoli* errors.

2.2 Kuril–Nedil errors

Kuril–Nedil errors involve replacing a short vowel character with its long counterpart or vice versa, sometimes yielding another valid word in the dictionary. For example, the word படம் (Transliteration: padam, Translation: picture) might be used instead of the word பாடம் (Transliteration: paadam, Translation: lesson). They are known as Kuril Nedil errors. Both படம் and பாடம் are valid words in the dictionary and might be used in an irrelevant context.

2.3 Mayangoli errors

Mayangoli errors arise from confusions among consonant graphemes associated with similar phonemes. The confusion groups in Tamil are as follows:

1. **'la' group:** ல/ள/ழ (Transliteration: la)
2. **'ra' group:** ர/ற (Transliteration: ra)
3. **'na' group:** ந/ன/ண (Transliteration: na)

These confusions also appear in UyirMei (consonant+vowel) combinations. A key practical challenge is that a Tamil writer might get confused and use them in irrelevant contexts. This work focuses exclusively on Mayangoli errors.

3 Related Work

In Tamil, spelling correction has been studied through dictionary-based, rule-based, statistical, and neural approaches. Dictionary-based methods remain effective for non-word errors but are weak for context-sensitive real-word errors where both candidates are valid words (Uthayamoorthy et al., 2019). Statistical approaches have been proposed for Tamil spelling correction (Sakuntharaj and Mahesan, 2016, 2017, 2018), typically modelling error likelihood and contextual probabilities; however, they often struggle with long-range dependencies and semantic appropriateness.

Neural methods have achieved strong results for context-aware correction, including transformer-based approaches and text-to-text models (Rajalakshmi et al., 2023; Elango and Pati, 2023). Yet, low-resource conditions for Tamil (limited task-specific annotated corpora) constrain model training and evaluation, and the lack of publicly available resources limits reproducibility. Multilingual pretrained encoder–decoder models such as mBART (Liu et al., 2020), mT5 (Xue et al., 2021), and NLLB (NLLBTeam et al., 2024) offer cross-lingual transfer that can benefit low-resource languages, and are well-suited for Seq2Seq transformations like spelling correction and text normalisation. Our work builds on these advances by providing a reproducible Mayangoli-specific pipeline and an empirical comparison of multilingual Seq2Seq models under controlled induction and cross-genre evaluation.

4 Data and Methodology

An overall pipeline of the proposed TamilMayangoliSpell framework is illustrated in Figure 1. The process starts with the raw Tamil corpora and is followed by abbreviation normalisation and text cleaning to avoid irregular sentence segmentation. Then, Mayangoli-specific substitution rules are used to produce linguistically plausible error patterns, constrained by dictionary membership. These are used to construct parallel sentence pairs for supervised training. Finally, multilingual Seq2Seq models are fine-tuned on the generated dataset and evaluated using both intrinsic metrics and cross-genre testing.

Algorithms 1–4 are used in the core pipeline for the creation of the parallel dataset. These algorithms are provided for clarity and reproducibility. The actual implementation of all the algorithms is available in the GitHub repository.[2]

[2] https://github.com/TamilGeekGirl/TamilMayangoliSpell/tree/main/Preprocessing%20scripts

4.1 Corpus source

TamilCorp (Yazhmozhi VM and Waller, 2025) is a balanced Tamil corpus comprising approximately 1.69 billion tokens collected from 17 genres, including newspapers, literature, government documents, and web content. The corpus is designed to ensure broad genre coverage—letters, newspapers, law, proverbs, essays, reviews, arts, magazines, finance, medicine, research journals, government, short stories, textbooks, other books, religious texts, and Wikipedia—drawing inspiration from established balanced corpora like Brown (Brown University, 2024) and the British National Corpus (British National Corpus, 2024). Data was obtained through web scraping and publicly available documents, with additional permissions secured from selected publishers, and collected in compliance with UK copyright regulations. A small chunk of the newspaper genre of TamilCorp (Yazhmozhi VM and Waller, 2025) was used as a text source for preprocessing and error induction. The resulting error induced parallel dataset is intended to represent context-sensitive errors where both erroneous and corrected forms are valid words.

4.2 Preprocessing

Preprocessing ensures that raw Tamil text is cleaned, normalised, and segmented into linguistically valid sentences. A major practical issue is sentence boundary detection. Abbreviations (e.g., month abbreviations ending with period, initials in names) can trigger false sentence splits. To remedy this, common abbreviations are normalised by expanding them prior to sentence splitting and cleaning. Algorithms 1 and 2 summarise the preprocessing pipeline: Algorithm 1 normalises Tamil text by expanding abbreviations and preventing incorrect sentence boundary detection, whereas Algorithm 2 implements memory-efficient chunk-based ingestion and preprocessing of large Tamil corpora into sentence-level units.

4.3 Error induction overview

Supervised training pairs are created by injecting Mayangoli errors into clean sentences:

$$\langle x_{\text{err}}, x_{\text{clean}} \rangle$$

For each word containing a target character (including UyirMei), candidate substitutions are generated within the same confusion group and only those where the substituted word exists in a dictionary (contains 2,583,001 words) are retained.[3] This ensures induced errors are context-sensitive real-word errors rather than non-words. Algorithms 3 and 4 summarise the Mayangoli error induction pipeline: Algorithm 3 constructs linguistically valid Mayangoli substitution mappings for all UyirMei characters and Algorithm 4 generates supervised context-sensitive Mayangoli error–correction sentence pairs using dictionary-constrained substitutions.

Algorithm 1 Abbreviation Normalisation

Require: Tamil text segment T, abbreviation map A
Ensure: Normalised text T'
1: $T' \leftarrow T$
2: **for all** $(abbr, full) \in A$ **do**
3: Replace occurrences of $abbr$ in T' with $full$
4: **end for**
5: Preserve multi-dot abbreviations to avoid sentence boundary errors
6: **return** T'

4.4 Advantages of the pipeline

The proposed pipeline has a few advantages for processing low-resource languages. They are as follows:

- It supports scalable dataset generation without manual annotation.
- Confusion groups are linguistically grounded so that the induced errors are linguistically plausible and context-sensitive.
- The modularity of our approach facilitates adopting this method to other types of errors in Tamil or another Dravidian language.
- The pipeline is combined with multilingual pretrained models to facilitate effective training with less supervised data.

4.5 Dataset size and split

The full pipeline yields 279,837 candidate pairs. We subsample 30K pairs to balance computational cost and training efficiency. We split this subset into 80/10/10 train/validation/test at the sentence-pair level to avoid overlap across splits.

[3] https://github.com/KaniyamFoundation/all_tamil_words/tree/master

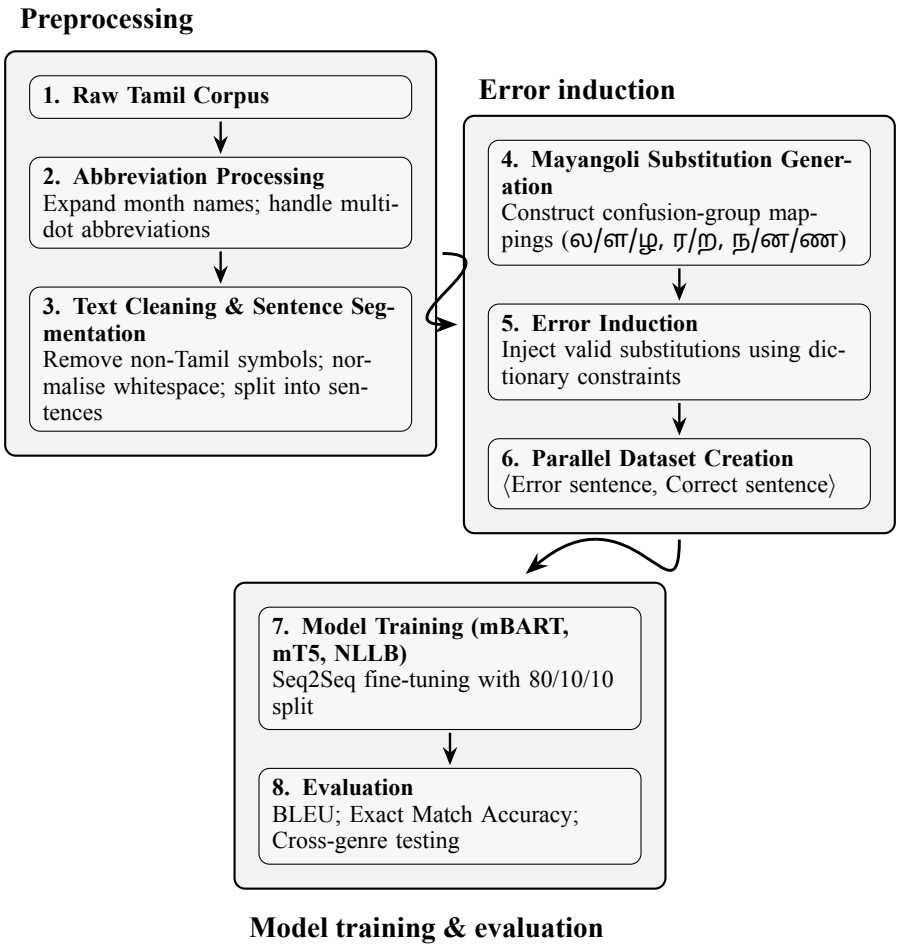

Figure 1: Overview of the TamilMayangoliSpell pipeline.

Algorithm 2 Sentence Extraction and Cleaning

Require: Raw Tamil corpus C, chunk size k
Ensure: List of cleaned sentences S

Initialise empty list S
while more text in C **do**
  Read next chunk c of size k
  $c \leftarrow \text{AbbreviationNormalisation}(c)$
  Remove non-Tamil symbols and normalise whitespace
  Segment c into sentences using Tamil-aware rules
  Append sentences to S
end while
return S

Algorithm 3 Mayangoli Substitution Generation

Require: Set of Tamil UyirMei characters U
Ensure: Substitution set R

Initialise empty set R
for all $u \in U$ **do**
  Decompose u into vowel and consonant components (v, c)
  if c belongs to a confusion group **then**
    for all $c' \neq c$ in same group **do**
      Generate $u' \leftarrow (v, c')$
      Add (u, u') to R
    end for
  end if
end for
return R

5 Models

Multilingual sequence-to-sequence (Seq2Seq) models are a natural choice for context-sensitive spelling correction, since they can be trained to map an incorrect sentence directly to its corrected form (Lewis et al., 2020; Liu et al., 2020). In this work, we focus on models that satisfy several practical requirements. Firstly, the model should include Tamil in its pretraining data so that its representations are compatible with the script (Conneau et al., 2020; Xue et al., 2021;

Algorithm 4 Error Induction and Dataset Construction

Require: Clean sentences S, dictionary D, substitutions R
Ensure: Dataset P of ⟨error, correct⟩ pairs

```
Initialise empty list P
for all s ∈ S do
    for all word w in s do
        for all (x, y) ∈ R do
            if x occurs in w then
                w′ ← replace x with y in w
                if w′ ∈ D then
                    Create s′ by replacing w with w′ in s
                    Add (s′, s) to P
                end if
            end if
        end for
    end for
end for
return P
```

NLLBTeam et al., 2024). Secondly, we restrict our choice to encoder–decoder architectures, as they are better suited for generating corrected text compared to models designed for token-level prediction (Lewis et al., 2020; Liu et al., 2020). Thirdly, we consider models that use subword tokenisation methods such as SentencePiece or Byte Pair Encoding (BPE), which are known to handle morphologically rich languages effectively (Kudo and Richardson, 2018; Sennrich et al., 2016). Finally, we prioritise models that have already shown strong performance in multilingual generation tasks, especially in low-resource settings (Liu et al., 2020; Xue et al., 2021; NLLBTeam et al., 2024).

In light of the above, we selected mBART (Liu et al., 2020), mT5 (Xue et al., 2021) and NLLB (NLLBTeam et al., 2024) for our experiments. The differences in pretraining objectives, vocabulary formation and handling of multilingual data makes them ideal for a comparative analysis with respect to Mayangoli error correction. We did not include models that use only the encoder like mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020), as they are intended for classification or mask prediction, so not well suited to sequence-to-sequence correction.

6 Experimental Setup

6.1 Training details

We fine-tuned all the models for 5 epochs using mixed precision (FP16) with early stopping (patience=2, based on validation loss) and saved the best checkpoint per run, ensuring convergence without overfitting. Validation loss stabilised within this range across models, consistent with prior findings on the efficient fine-tuning of pretrained Seq2Seq models (Popel and Bojar, 2018; Dodge et al., 2020; Mosbach et al., 2020). Hyperparameters are selected via a small grid:

- Batch size: $\{4, 8\}$
- Learning rate: $\{3 \times 10^{-5}, 1 \times 10^{-4}\}$

These parameters were selected because they are consistently reported as the chief factors affecting neural sequence-to-sequence optimisation dynamics and model performance (Smith, 2018; Popel and Bojar, 2018; Mosbach et al., 2020). Other hyperparameters (e.g., dropout, weight decay, and warm-up steps) were set to their default values as provided by the underlying framework, since relevant literature indicates that they have relatively smaller effects on convergence and generalisation (Devlin et al., 2019; Liu et al., 2020).

6.2 Decoding

During inference, greedy decoding is used with a maximum sequence length of 128 for all models to ensure consistent decoding conditions.

6.3 Metrics

We report:

- **BLEU** (Papineni et al., 2002), which measures n-gram overlap.
- **Exact Match Accuracy (EMA)** (Rajpurkar et al., 2016), a strict metric requiring full-string equality.

EMA is most suited for spelling correction tasks, as it is stricter; even an error in a single character invalidates the correction. However, BLEU is also selected as an evaluation metric to capture partial correctness and to gain insight into the behaviour of the model when there are no exact matches.

7 Results

7.1 Main results on newspaper data

Table 1 summarises validation performance for each model under the hyperparameter grid. Overall, mT5 performs the best, achieving BLEU 99.28 and EMA 93.50%. mBART is stable across hyperparameter choices but lags in peak accuracy. NLLB achieves competitive BLEU but yields an EMA of 0 across all configurations.

mBART exhibits identical performance across hyperparameter configurations, suggesting stable convergence under the limited hyperparameter grid and dataset size (30K). This indicates low sensitivity to optimisation settings in this regime, consistent with observations in low-resource fine-tuning of pretrained Seq2Seq models (Dodge et al., 2020; Mosbach et al., 2020).

7.2 Cross-genre evaluation

As mT5 achieved the best overall performance (in newspaper genre), we tested its generalisation ability by evaluating it on a different genre (short stories; 30K) using the same hyperparameter grid. Results in Table 2 show consistently high BLEU ($\geq$ 98.87) and EMA ($\geq$ 89.60), with peak BLEU 99.28 and peak EMA 93.50.

Cross-genre evaluation yields identical scores to in-domain validation across all configurations. We verified that the datasets are disjoint and that no data leakage occurs. This behaviour may be attributed to the controlled error induction process, which produces similar distributions of Mayangoli confusions across genres, resulting in consistent task difficulty.

The cross-genre evaluation suggests that the learned correction behaviour generalises beyond the training domain, although real-world deployment may involve noisier and more diverse error patterns.

8 Analysis and Discussion

While the underlying models are standard, our contribution lies in a linguistically grounded error induction framework for Tamil orthography. Unlike generic noise injection, we enforce phonologically valid, dictionary-constrained substitutions to generate context-sensitive real-word errors. We intentionally use standard pretrained Seq2Seq architectures to isolate the effects of data generation and task formulation rather than architectural variation.

Although evaluation relies on synthetic induction, substitutions are constrained by phonological confusion groups and dictionary validity, producing linguistically plausible context-sensitive real-word errors. Unlike dictionary-based or surface-form approaches, Seq2Seq models capture sentence-level contextual dependencies required for resolving context-sensitive real-word Mayangoli confusions.

8.1 Why does mT5 perform best?

We attribute mT5's advantage to (i) its uniform text-to-text objective that naturally matches correction as "translate erroneous $\rightarrow$ correct", and (ii) SentencePiece tokenisation that often better handles subword composition in morphologically rich scripts. Empirically, mT5 exhibits both high BLEU and high EMA, indicating that improvements are not merely n-gram overlap but full-sequence correctness.

8.2 NLLB: high BLEU, zero EMA

NLLB yields a non-trivial BLEU yet a 0 EMA, indicating that the outputs share partial n-gram overlap but fail to achieve exact correction. We observed this pattern consistently across hyperparameter configurations and decoding settings (greedy decoding, max_length=128). A plausible explanation is that the model produces near-miss variants (e.g., correct context but wrong character choice or minor token boundary artifacts) that preserve many n-grams but fail strict equality. This highlights that BLEU can be misleading for orthographic correction and motivates stricter and character-level metrics.

8.3 Additional evaluation considerations

EMA is strict and may under-credit partial improvements (e.g., correcting one of multiple errors). Conversely, BLEU may over-credit near-misses. In future work, character-level F-score, edit distance, minimal-edit accuracy, and human evaluation could provide more nuanced conclusions for end-user applications.

9 Deployment and Practical Usage

The trained Mayangoli error correction models can be used as an end-user application to help Tamil writers and people with language difficulties, such as dyslexia. The modular architecture of the system enables future work to integrate it easily into web-based or mobile applications. We provide two

Table 1: Validation performance of multilingual Seq2Seq models on the newspaper dataset (30K sentence pairs). Best configuration per model is highlighted in bold.

Model	Batch	Epochs	LR	BLEU ↑	Exact Match (%) ↑
mBART	4	5	3×10^{-5}	**92.98**	**89.53**
	4	5	1×10^{-4}	**92.98**	**89.53**
	8	5	3×10^{-5}	**92.98**	**89.53**
	8	5	1×10^{-4}	**92.98**	**89.53**
mT5	4	5	3×10^{-5}	98.96	90.52
	8	5	3×10^{-5}	98.87	89.60
	4	5	1×10^{-4}	99.27	**93.50**
	8	5	1×10^{-4}	**99.28**	93.38
NLLB	4	5	3×10^{-5}	93.04	0.00
	8	5	3×10^{-5}	**94.73**	0.00
	4	5	1×10^{-4}	90.77	0.00
	8	5	1×10^{-4}	93.63	0.00

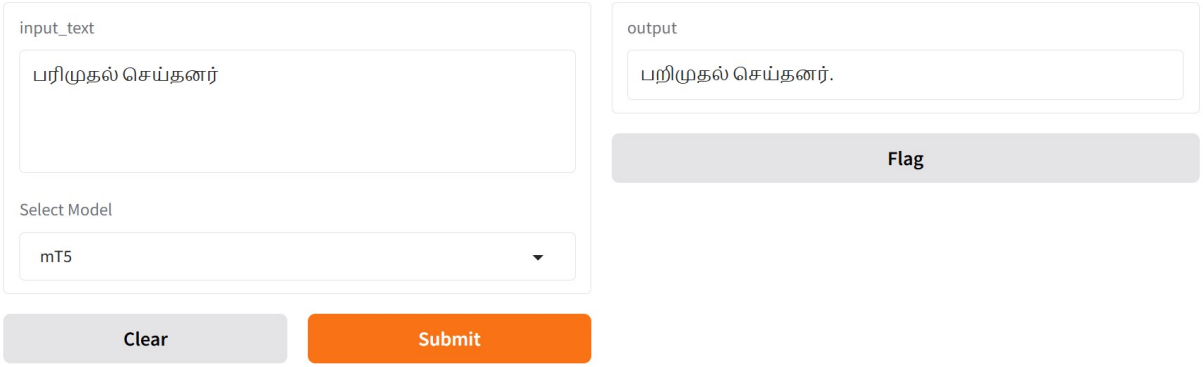

Figure 2: GUI-based Mayangoli error correction (mT5).

Table 2: Cross-genre evaluation of mT5 on Tamil short stories. Best scores are highlighted in bold.

BS	Ep	LR	BLEU ↑	EMA ↑
4	5	3×10^{-5}	98.96	90.52
8	5	3×10^{-5}	98.87	89.60
4	5	1×10^{-4}	99.27	**93.50**
8	5	1×10^{-4}	**99.28**	93.38

lightweight local deployment interfaces. The CLI is beneficial for developers and for integration into larger NLP pipelines. The GUI is beneficial for non-technical users. It is also well-suited for educational tools and writing aids.

Figures 2 and 3 illustrate the GUI-based deployment of Mayangoli error correction using the best-performing mT5 and mBART models. Figure 2 demonstrates the correction of a shorter sentence with two words, whereas Figure 3 demonstrates the correction of a longer sentence with five words. Additional deployment screenshots are available in the GitHub repository.

10 Limitations

Scope limitation. We focus exclusively on *Mayangoli* confusions and do not address other Tamil spelling error classes such as Kuril–Nedil errors, sandhi-related orthographic changes, non-word errors, or code-mixed writing. The system should therefore not be interpreted as a complete Tamil spell-checking solution.

Synthetic data limitation. Training and evaluation rely on synthetic error induction. Although we constrain substitutions by dictionary validity to ensure context-sensitive real-word errors, the induced distribution may not be fully representative of naturally occurring user errors (e.g., frequency of confusions, multi-error sentences, informal spelling, dialectal variation). As a result, reported scores may be optimistic relative to real-world text.

Compute and search limitation. Due to com-

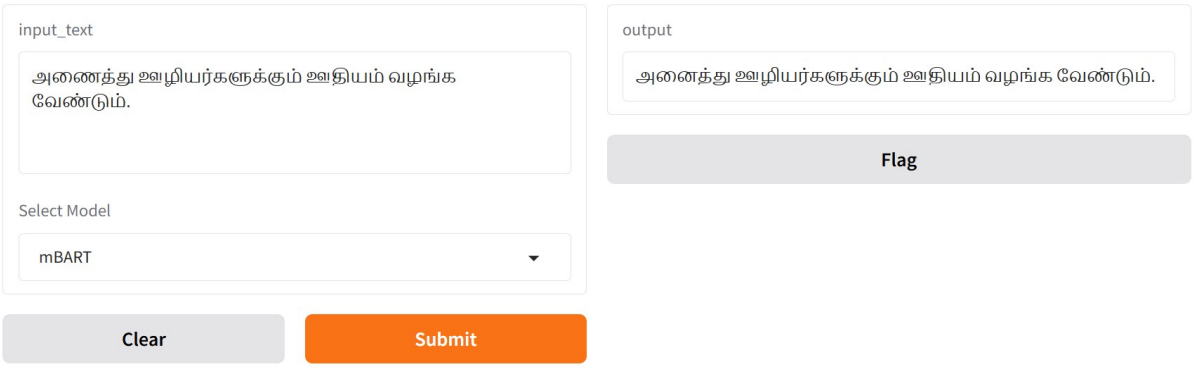

Figure 3: GUI-based Mayangoli error correction (mBART).

putational constraints, we train on a subsample of 30,000 sentence pairs and explore only a limited hyperparameter grid. Larger-scale fine-tuning using the full **TamilCorp** dataset (Yazhmozhi VM and Waller, 2025), additional model sizes, or broader optimisation strategies may change absolute performance and potentially alter model rankings.

Metric limitation. We evaluate using BLEU and EMA, which do not fully capture user-centric utility (partial corrections, edit distance reduction, or acceptability when multiple corrections are plausible). Metrics such as character error rate (CER) and edit distance are more suitable for orthographic correction and will be incorporated in future work.

Decoding limitation. We use a fixed decoding configuration (greedy decoding, max_length=128) for consistency. Alternative decoding settings may affect exact-match performance, especially for models sensitive to generation dynamics.

11 Conclusion

This paper explores how multilingual sequence-to-sequence (Seq2Seq) models can be used to correct *Mayangoli errors*, which are a category of context-sensitive errors in Tamil. We fine-tune three models—mBART, mT5, and NLLB—on a dataset of 30,000 sentence pairs derived from the newspaper genre of TamilCorp, a balanced Tamil corpus. To support this, we design a preprocessing and error-induction pipeline that generates linguistically plausible Mayangoli error-correction pairs.

Across experiments, all three models achieve strong BLEU scores under different hyperparameter settings. Among them, mT5 performs the best, achieving a BLEU score of 99.28 and an Exact Match Accuracy (EMA) of 93.50% (a batch size of 8, learning rate of 1×10^{-4}, and 5 training epochs). Performance of mBART over various hyperparameter settings is stable (BLEU: 92.98, EMA: 89.53%). This indicates that mBART may be more robust to hyperparameter tuning. However, it does not reach the same peak performance as mT5. NLLB generates strong BLEU scores but does not achieve exact match accuracy. This suggests high n-gram overlap does not guarantee correct full-sequence prediction.

Overall, the findings demonstrate that mT5 performs strongly for Mayangoli error correction in Tamil. This may result from its text-to-text formulation and subword tokenization capturing Tamil's morphological variation effectively. More generally, these results reveal that multilingual Seq2Seq models can be powerful tools for correcting context-sensitive spelling errors in Tamil. Future work involves generalising this approach for other error types in Tamil and evaluating performance in real-world usage settings. With more domain adaptation and user-based evaluation, **TamilMayangoliSpell** could substantially improve the accuracy and usability of Tamil language technology.

References

British National Corpus. 2024. Bnc homepage. http://www.natcorp.ox.ac.uk/. Accessed: 2024-10-20.

Brown University. 2024. Brown corpus manual. http://korpus.uib.no/icame/manuals/BROWN/INDEX.HTM. Accessed: 2024-10-20.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco

Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8440–8451.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Vaan Amuthu Elango and Peeta Basa Pati. 2023. Tamil text error correction with multi-lingual t5 model. In *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)*, pages 1–6.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2020. On the stability of fine-tuning BERT: misconceptions, explanations, and strong baselines. *CoRR*, abs/2006.04884.

NLLBTeam, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2024. Scaling neural machine translation to 200 languages. *Nature*, 630(26):841–846.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.

K.R. Pillai. 1975. *Tolkappiyam: Text and Commentary*. University of Madras Press, Chennai, India.

Martin Popel and Ondřej Bojar. 2018. Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

Ratnavel Rajalakshmi, Varsha Sharma, and Anand Kumar M. 2023. Context sensitive tamil language spellchecker using roberta. In *Speech and Language Technologies for Low-Resource Languages*, pages 51–61, Cham. Springer International Publishing.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Ratnasingam Sakuntharaj and Sinnathamby Mahesan. 2016. A novel hybrid approach to detect and correct spelling in tamil text. In *2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*, pages 1–6. IEEE.

Ratnasingam Sakuntharaj and Sinnathamby Mahesan. 2017. Use of a novel hash-table for speeding-up suggestions for misspelt tamil words. In *2017 IEEE International Conference on Industrial and Information Systems (ICIIS)*, pages 1–5. IEEE.

Ratnasingam Sakuntharaj and Sinnathamby Mahesan. 2018. Detecting and correcting real-word errors in tamil sentences. *Ruhuna Journal of Science*, 9(2):150–159.

Anbukkarasi Sampath and Varadhaganapathy Shanmugavel. 2023. Hybrid tamil spell checker with combined character splitting. *Concurrency and Computation: Practice and Experience*, 35(1):e7440.

Jananie Segar and KengatharaiyerSarveswaran. 2015. Contextual spell checking for tamil language. In *14th Tamil Internet 2015 Conference, Singapore.*, pages 379–383.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Lakshikka Sithamparanathan and Thayasivam Uthayasanker. 2019. A sinhala and tamil extension to generic environment for context-aware correction. In *2019 National Information Technology Conference (NITC)*, pages 102–106. IEEE.

Leslie N. Smith. 2018. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*.

Keerthana Uthayamoorthy, Kirshika Kanthasamy, Thavarasa Senthaalan, Kengatharaiyer Sarveswaran, and Gihan Dias. 2019. Ddspell - a data driven spell checker and suggestion generator for the tamil language. In *2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer)*, pages 1–6. IEEE.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498.

Yazhmozhi VM and Annalu Waller. 2025. Building a balanced tamil corpus: Eda and lexical diversity comparison with english. In *2025 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)*, pages 1–6.