DLRG at BHASHA: Task 1 (IndicGEC): A Hybrid Neurosymbolic Approach for Tamil and Malayalam Grammatical Error Correction

Akshay Ramesh and Ratnavel Rajalakshmi
School of Computer Science and Engineering
Vellore Institute of Technology, Chennai
E-mail : rajalakshmi.r@vit.ac.in

Abstract

Grammatical Error Correction (GEC) for low-resource Indic languages remains challenging due to limited annotated data and morphological complexity. We present a hybrid neurosymbolic GEC system that combines neural sequence-to-sequence models with explicit language-specific rule-based pattern matching. Our approach leverages parameter-efficient LoRA adaptation on aggressively augmented data to fine-tune pre-trained mT5 models, followed by learned correction rules through intelligent ensemble strategies. The proposed hybrid architecture achieved 85.34% GLEU for Tamil (Rank 8) and 95.06% GLEU for Malayalam (Rank 2) on the provided IndicGEC test sets, outperforming individual neural and rule-based approaches. The system incorporates conservative safety mechanisms to prevent catastrophic deletions and over-corrections, thus ensuring robustness and real-world applicability. Our work demonstrates that extremely low-resource GEC can be effectively addressed by combining neural generalization with symbolic precision.

1 Introduction

Grammatical Error Correction (GEC) focuses on automatically detecting and correcting errors in written text, including spelling mistakes, grammatical inconsistencies, punctuation errors, and word choice issues. There has been substantial research progressing with state-of-the-art results for high-resource languages like English (Bryant et al., 2019; Grundkiewicz et al., 2019). However, GEC for Indic languages face severe challenges from limited annotated sentence pairs, rich inflectional morphology characteristic of agglutinative languages, and unique Indic script properties regarding Unicode representation and normalization.

Tamil and Malayalam, Dravidian languages with over 75 million and 38 million speakers respectively, exemplify these challenges. The shared IndicGEC datasets exhibit extremely low-resource settings, necessitating novel approaches beyond standard neural fine-tuning (Bryant et al., 2019). We present a hybrid neurosymbolic architecture strategically combining neural and symbolic approaches. The neural component provides generalization to unseen error patterns through mT5 models that are fine-tuned with LoRA (Xue et al., 2021; Hu et al., 2021), while the symbolic component ensures high precision on known error patterns through explicit rule extraction from training data. The core innovation lies in the intelligent ensemble that selectively applies exact matches, neural predictions, or rule-based corrections based on input characteristics and multiple safety validation mechanisms.

The key contributions of our work are:

- A Novel hybrid architecture that combines neural sequence-to-sequence models with pattern-based corrections for low-resource GEC with conservative safety mechanisms.
- Language-specific data augmentation strategy that generates up to 10,000 synthetic examples from limited gold pairs using morphology-aware noise injection.
- Robust Ensemble selection mechanism with safety thresholds to prevent catastrophic deletions, over-corrections, and output degeneration.

The system successfully generated corrections for test inputs, demonstrating that the hybrid approach effectively leverages limited training data while prioritizing computational efficiency, the preservation of output quality, and real-world deployment through conservative correction strategies.

2 Related Works

2.1 Grammatical Error Correction

Recent GEC research, especially for English, has been dominated by neural approaches where Transformer-based models and large pre-trained models like BART and T5 achieved state-of-the-art results (Zhao et al., 2019; Kaneko et al., 2020; Katsumata and Komachi, 2020; Rothe et al., 2021). However, these approaches require substantial training data, say millions of examples and computational resources. In low-resource languages like Tamil, there are few works that focus on spelling errors and correction (Rajalakshmi, R et al.,), but grammatical error correction is not explored much. Low-resource GEC remains challenging, with researchers exploring synthetic data generation for Czech GEC (Naplava and Straka, 2019) and feedback comment generation for low-resource languages (Flachs et al., 2021). Our work differs from these by combining neural and symbolic approaches with explicit safety mechanisms, specifically, for extremely low-resource settings.

2.2 Multilingual and Indic Language GEC

Multilingual models such as mBART and mT5 exhibit promising potential for cross-lingual transfer (Liu et al., 2020; Xue et al., 2021). Complementing this, Rothe et al. (2021) demonstrated that mT5 fine-tuning can achieve competitive GEC performance. However, direct application to Indic languages with minimal data remains unexplored. GEC for Indic languages is nascent, with most prior work focusing primarily on spell-checking rather than on comprehensive grammatical correction (Joshi et al., 2012). The shared IndicGEC tasks represent one of the first systematic efforts in this area. Our model is one among the first to address Dravidian languages with a modern neural-symbolic hybrid method incorporating robustness mechanisms.

2.3 Hybrid NLP Systems

The neurosymbolic approach combines neural learning with symbolic reasoning. Recent works in this area, include neural symbolic parsers, hybrid question answering, and rule-augmented neural models (Platanios et al., 2021; Mitra and Baral, 2016). For GEC specifically, Awasthi et al. (2019) combined neural models with rule-based post-editing for English. On the other hand, our work extends hybrid methods to extremely low-resource scenarios with explicit safety validation, demonstrating that explicit pattern extraction from minimal training data combined with neural generalization and conservative acceptance criteria can achieve superior performance while preventing common failure modes.

2.4 Parameter-Efficient Fine-Tuning

Hu et al. (2021) demonstrated that LoRA (Low-Rank Adaptation) enables efficient fine-tuning by injecting trainable low-rank matrices into frozen pre-trained models, reducing trainable parameters by over 99% while maintaining performance. Our work leverages LoRA to fine-tune mT5-base and mT5-small for Tamil and Malayalam GEC, respectively, with limited training examples, enabling effective adaptation and preventing overfitting.

2.5 Language Model Selection for Sequence-to-Sequence GEC

While monolingual BERT-based encoder models exist for both Tamil (l3cube-pune/tamil-bert) and Malayalam (l3cube-pune/malayalam-bert), these models are fundamentally unsuitable for GEC tasks due to their encoder-only architecture. GEC is inherently a sequence-to-sequence task requiring both encoding input sentences and generating corrected outputs, necessitating encoder-decoder architectures like T5 or BART.

BERT-based models, being encoder-only, can only produce contextual representations and are designed for classification, token labelling, or extraction tasks rather than text generation. Adapting BERT for generation would require adding a decoder component from scratch, essentially reconstructing an encoder-decoder model without the benefits of pre-trained generation capabilities. Furthermore, no production-ready monolingual T5-style encoder-decoder models exist for Tamil or Malayalam in public repositories. While researchers have created language-specific adaptations by pruning multilingual models (e.g., Russian T5), similar efforts for Dravidian languages remain unpublished or unavailable.

Therefore, we leverage mT5, a multilingual T5 variant pre-trained on 101 languages including Tamil and Malayalam, which provides the necessary encoder-decoder architecture for GEC while offering cross-lingual transfer benefits from related languages. The mT5 family's availability in multiple sizes (small, base, large) enables capacity-driven design choices suitable for our low-resource

setting, as demonstrated in our ablation studies (Section 4.4).

3 System Architecture

We present differentiated frameworks for Tamil and Malayalam GEC, reflecting language-specific requirements. Figures 1 & 2 illustrate the complete system workflows for Tamil and Malayalam languages respectively. This differentiation reflects Tamil's morphological complexity, which requires greater model capacity, and Malayalam's higher observed risk of neural over-correction, requiring conservative safety mechanisms.

3.1 Tamil GEC Architecture

The Tamil system employs a five-stage hierarchical pipeline that combines neural and symbolic approaches strategically. First, marker extraction isolates formatting elements (-, ;-) from core content using regex patterns, enabling focus on linguistic content. Second, rule-based priority checking matches queries against sentence templates and training data; exact matches return stored corrections immediately. Third, neural generation uses mT5-base with LoRA adaptation, employing beam search when no exact match exists. Fourth, pattern enhancement applies 25+ manually curated Tamil error patterns. Finally, marker reattachment deterministically restores original formatting.

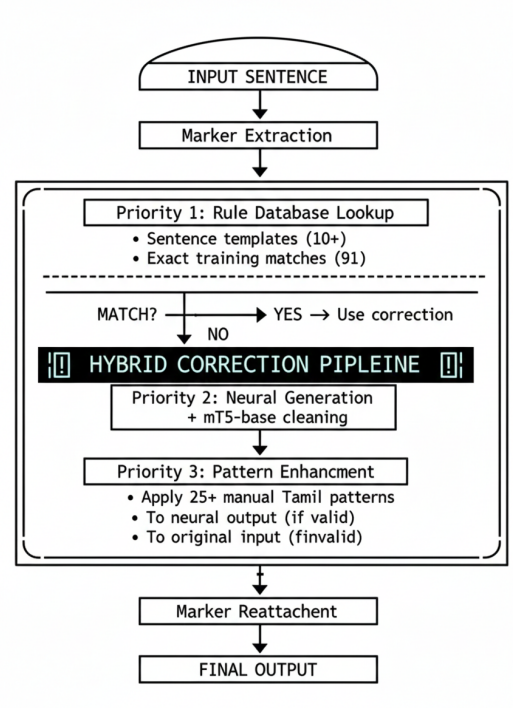

Figure 1: Architecture Diagram for Tamil GEC

The innovative hierarchical correction strategy operates at three junctures: pre-neural exact matching for high-confidence corrections, post-neural pattern application to enhance valid generations, and rule-only fallback with similarity matching for invalid outputs. The four tiers include: (1) exact sentence matches from templates, (2) training data matches offering perfect accuracy for 91 pairs, (3) enhanced neural generation with pattern enhancement for valid outputs, and (4) rule-only fallback for truncated, degenerate, or empty outputs using similarity-based matching.

The following Tamil example illustrates the correction process:

1. **Input:** thozhilsaalai iyandhithithin sattham (தொழிற்சாலை இயந்தித்தின் சத்தம், "factory mashine's noise").
2. **Tier 1 (Rule Lookup):** No template match.
3. **Tier 2 (Exact Match):** No training match.
4. **Tier 3 (Neural Generation):** The input is passed to the neural model.
5. **Pattern Enhancement:** This step detects the morphological error iyandhithith (இயந்தித்). A manual rule is applied: iyandhithith → iyanthira (இயந்திர).
6. **Output:** thozhilsaalai iyanthirathin sattham (தொழிற்சாலை இயந்திரத்தின் சத்தம், "factory machine's noise").

3.2 Malayalam GEC Architecture

The Malayalam system employs conservative parallel processing pipeline with safety-first ensemble selection. The workflow begins with exact match checking that validates inputs against learned corrections. If matched, the system returns the stored correction immediately, bypassing neural generation. When no exact match exists, the system proceeds to parallel processing where neural generation using mT5-small with LoRA and rule-based candidate preparation occur simultaneously. The neural output then undergoes comprehensive safety validation checking Malayalam character presence, token overlap ratios, length ratios, and deletion thresholds. Based on validation results and similarity analysis, the ensemble selector chooses between the neural output, the rule-based candidate, or falls back to the original input.

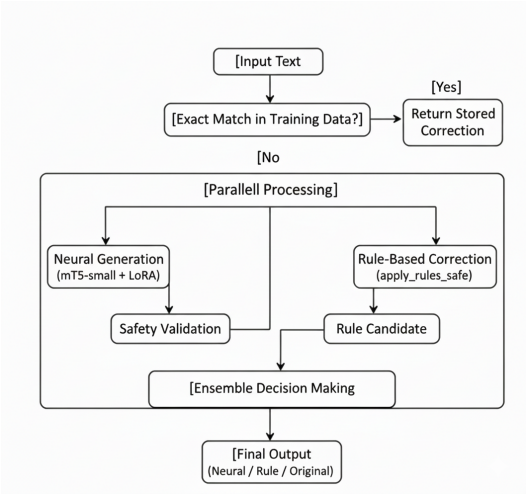

Figure 2: Architecture diagram for Malayalam

The key innovation of Malayalam architecture lies in its parallel processing, combined with conservative selection, rather than sequential transformation. Both neural and rule-based components process input independently, with final decisions made through confidence-based ensemble selection with multi-layered safety validation. The safety mechanisms validate neural outputs against multiple criteria: Malayalam character presence ($\geq$1 character in U+0D00-U+0D7F), token overlap ($\geq$45% Jaccard similarity), length ratio ($\geq$50% preservation unless overlap >90%), and deletion ratio ($\geq$45% unless overlap >90%).

The following example illustrates the parallel processing and ensemble decision-making within the Malayalam pipeline:

1. **Input:** vaakanam odichchu (വാകണം ഓടിച്ചു, "vehikle drove").

2. **Initial Check:** No exact match found in training data.

3. **Parallel Processing:**

   - **Neural Path:** Generates vaahanam odichchu (വാഹനം ഓടിച്ചു, "vehicle drove").
   - **Rule-Based Path:** Identifies the pattern: വാകണം (vaakanam) $\rightarrow$ വാഹനം (vaahanam).

4. **Safety Validation:** The neural output passes all checks (e.g., Malayalam chars $\geq$, 50% token overlap $\geq$, 100% length $\geq$, 0% deletion $\geq$).

Table 1: Neural component configuration for Tamil and Malayalam GEC (Grammatical Error Correction).

Component	**Tamil GEC**	**Malayalam GEC**
Base Model	mT5–base (580M Parameters, 12 layers, 768 hidden dims, 8 heads)	mT5–small (300M Parameters, 8 layers, 512 hidden dims, 6 attention heads)
LoRA Rank (r)	16	8
LoRA Alpha (α)	32	16
Target Modules	Q, V, K, O	Q, V Augmented
Dropout	0.1	0.1
Corpus Size	5000 examples	10000 examples
Generation Strategy	Beam Search (Width 6, LP 0.8) and repetition penalty (1.1)	Conservative Beam search (LP 1.0, Repetition P. 1.2, no-repeat-n-gram size 3)
Augmentation Focus	Vowel dropping, Char Duplication, Punctuation, Perturbation	Vowel sign dropping, Chillu variation, Punctuation normalization

5. **Ensemble Selection:** The system selects the neural output.

6. **Output:** vaahanam odichchu (വാഹനം ഓടിച്ചു, "vehicle drove").

3.3 Neural Component Design

Both systems use a pre-trained multilingual mT5 model, adapted using LoRA on aggressively augmented data with carefully chosen base model capacities. Table 1 summarizes the configurations.

Base Model Selection Rationale: Tamil exhibits highly complex agglutinative morphology with extensive case marking (8 cases) and verb conjugations requiring substantial model capacity to capture morphological patterns. The larger mT5-base (580M parameters, 12 layers) with higher LoRA rank (16) provides sufficient representational capacity for Tamil's morphological complexity without extreme overfitting, prioritizing correction coverage for diverse error patterns. Conversely, Malayalam, while also agglutinative, presents a higher risk of neural over-correction in our constrained dataset due to observed generation instability during preliminary experiments. The smaller mT5-small (300M parameters, 8 layers) with conservative LoRA rank (8) reduces overfit-

ting risk and generation volatility, prioritizing output reliability and stability reinforced by stringent ensemble safety gates. This differentiation reflects empirical findings that Tamil benefits from capacity while Malayalam requires conservative generation.

3.4 Data Augmentation

Data augmentation strategies were designed specifically for each language to mitigate data scarcity through controlled noise injection. For Tamil, augmentation included vowel dropping targeting 12 Tamil vowels (a, aa, i, ii, u, uu, e, ee, o, oo, ai, au /அ, ஆ, இ, ஈ, உ, ஊ, எ, ஏ, ஒ, ஓ, ஐ, ஔ), character duplication and deletion, punctuation perturbation, and word order shuffling, expanding from 91 to 5,000 examples representing a 55-fold increase. For Malayalam, augmentation focused on vowel sign dropping targeting 12 signs (-aa, -i, -ii, -u, -uu, -ri, -e, -ee, -ai, -o, -oo, -au / ാ, ി, ീ, ു, ൂ, ൃ, െ, േ, ൈ, ൊ, ോ, ൌ), safe character duplication and deletion avoiding the first two characters to prevent catastrophic truncation, adjacent word swapping excluding the first word to maintain sentence structure, comma spacing removal, punctuation normalization, and chillu variation handling modern-traditional pairs (ṉ/n-virama, ṉ/ṉ-virama, ḷ/l-virama, ḷ/ḷ-virama, ṛ/r-virama, ḳ/k-virama / ൻ/ന്, ൺ/ണ്, ൽ/ല്, ൾ/ള്, ർ/ര്, ൿ/ക്). The Malayalam augmentation process included similarity filtering, maintaining values between 0.6 and 0.98, and length preservation checks requiring at least 50% of the original length, expanding the corpus to 10,000 examples. Each original sentence underwent one or two random transformations, significantly enhancing model robustness while preventing spurious noise pattern learning.

3.5 Training Configuration

Training configuration remained consistent across both languages. Both systems employed AdamW optimizer under FP16 precision with a learning rate of 3e-4, effective batch size of 8, weight decay of 0.01, and training for 10 epochs with early stopping enabled to prevent overfitting. This configuration balanced training efficiency with model quality for extremely low-resource settings.

3.6 Rule-based and Ensemble Components

The symbolic component provides language-specific error pattern handling with deterministic high-precision corrections. The Tamil rule-based system stores all 91 training input-output pairs as exact sentence matches for perfect precision. It incorporates over 25 manually curated domain patterns covering common orthographic errors such as The symbolic component provides language-specific error pattern handling with deterministic high-precision corrections. The Tamil rule-based system stores all 91 training input-output pairs as exact sentence matches for perfect precision. It incorporates over 25 manually curated domain patterns covering common:

- **Orthographic errors** such as vaakanam → vaahanam, kalluurari → kallūri (வாகணம் → வாகனம், கல்லூரி → கல்லூரி).
- **Vowel lengthening errors:** (thoon → thūn, thookkam → tūkkam / தூண் → தூண், தூக்கம் → தூக்கம்).
- **Consonant errors** (iyandhithith → iyanthira, saramangal → siramangal / இயந்தித் → இயந்திர, சரமங்கள் → சிரமங்கள்).
- **Common word corrections** (haaran → haarn, ventum → ventum / ஹாரன் → ஹார்ன், வேன்டும் → வேண்டும்).

Additionally, the system maintains over 10 full sentence templates for frequent multi-error patterns and employs similarity matching using SequenceMatcher with a 0.75 threshold for approximate matches.

The Malayalam rule-based system stores all training pairs as exact sentence matches for immediate high-confidence corrections. It implements automated phrase-level pattern learning where SequenceMatcher identifies phrase replacements up to 6 tokens from training data. Safe phrase replacement employs regex word-boundary matching to prevent word fragmentation with validation ensuring preservation of at least the minimum required tokens, avoiding unexpected first character changes, preventing text from beginning with punctuation, and limiting application to one replacement per sentence to avoid cascading errors. The system also performs punctuation normalization by removing trailing commas and quotes, normalizing spacing, and collapsing whitespace.

The ensemble layer integrates neural and symbolic outputs using differentiated strategies. The

Tamil ensemble employs a confidence-driven hierarchical approach prioritizing exact matches from sentence-level and training data matches, followed by neural predictions refined through rule-based post-processing, and finally rule-only fallbacks for unseen cases. Post-processing cleans artifacts including <extra_id> tokens and "correct:" prefixes, normalizes whitespace, and applies punctuation corrections.

The Malayalam ensemble employs a safety-first parallel selection strategy. Exact matches are prioritized with training data lookups providing perfect accuracy. Validated neural predictions must pass all safety criteria, including character presence, $\geq$45% token overlap, $\geq$50% length ratio, and $\leq$45% deletion ratio. Rule-based enhancements are applied to valid neural outputs through safe phrase replacement. Similarity-based selection chooses between neural and rule candidates based on overlap with the original input. When both approaches produce high-similarity outputs exceeding 95% for rules or 88% for neural comparisons, the system conservatively prefers candidates maintaining higher token overlap. Fallback to the original input occurs when both candidates fail safety checks or when neural generation produces empty or severely truncated outputs. The ensemble strategy explicitly tracks usage statistics including neural used, rule used, exact used, and fallback used, providing transparency in the correction decision process.

4 Experiments and Results

4.1 Experimental Setup

The IndicGEC training sets contain sentence pairs in CSV format with input and output sentence columns. Test data was provided without gold standard outputs, simulating real-world deployment where systems generate corrections independently. This blind evaluation assesses ability to handle diverse error patterns without reference targets. Tamil dataset includes 91 training pairs augmented to 5,000, with 16 validation pairs and 65 test inputs. Malayalam dataset includes limited training pairs augmented to 10,000, with validation set available and 102 test inputs. GLEU served as the primary evaluation metric balancing n-gram precision and recall.

Three configurations were compared for both languages:

1. Neural-only using mT5-base (for Tamil, r=16) or mT5-small (for Malayalam, r=8) fine-tuned on augmented data

2. Rule-only employing multi-layer pattern matching using exact sentences, domain patterns, phrase-level corrections, and similarity-based matching with threshold 0.75

3. The proposed hybrid ensemble combining neural predictions with rule-based processing, marker preservation for Tamil, and safety validation for Malayalam.

Implementation used Hugging Face Transformers v4.35, PEFT v0.7, and PyTorch. The Tamil system was fine-tuned for 10 epochs on 5,000 augmented examples with 91 exact corrections, over 25 manually curated patterns, and over 10 sentence templates, using regex-based marker extraction and reattachment for formatting integrity. The Malayalam system was fine-tuned for 10 epochs on 10,000 augmented examples with automated phrase-level pattern extraction via SequenceMatcher and safety validation thresholds requiring at least 1 Malayalam character, at least 45% token overlap, at least 50% length ratio, and at most 45% deletion ratio, with ensemble similarity thresholds of 95% for rule-based and 88% for neural comparisons.

4.2 Results

On validation sets, Tamil achieved 80.47% GLEU (16 examples) while Malayalam achieved 55.21% GLEU.

On blind test sets, Tamil achieved 85.34% GLEU (65 inputs securing overall Rank 8), while Malayalam achieved 95.06% GLEU (102 inputs securing impressive Rank 2). Both hybrid models significantly outperformed individual neural-only and rule-only baselines, demonstrating the effectiveness of the neurosymbolic approach for extremely low-resource GEC.

4.3 Error Analysis

Error analysis on test sets revealed the systems' capabilities across diverse error types. Representative examples are provided in Table 2 for both Tamil and Malayalam.

The Tamil system demonstrated capability handling morphological complexity, including transformations like iyandhithith → iyanthira /

இயந்தித் → இயந்திர, multi-token corrections such as haaran → haarn / ஹாரன் → ஹார்ன் and vaakanam → vaahanam / வாகணம் → வாகனம் simultaneously, verb form corrections with subject-verb agreement, vowel length normalization converting -uaa → -ū / ாா → ா, word order reordering while preserving markers, handling multiple simultaneous errors, and punctuation insertion.

The Malayalam system demonstrated spelling corrections, conservative preservation when no correction was needed, and token-level preservation. Safety validation mechanisms successfully prevented catastrophic deletions and overcorrections, maintaining input integrity when corrections were uncertain.

4.4 Ablation Study: Model Capacity Analysis

To validate our language-specific model selection and to address the impact of model capacity on performance, we conducted ablation study by swapping mT5 variants between languages. Specifically, we trained the Tamil GEC with mT5-small (originally used mT5-base) and Malayalam GEC with mT5-base (originally used mT5-small), maintaining identical training configurations, augmentation strategies, and ensemble mechanisms.

The ablation results strongly validate our differentiated model selection strategy. For Tamil, reducing model capacity from mT5-base to mT5-small resulted in a 5.30 percentage point drop in validation GLEU (from 80.47% to 75.17%), demonstrating that Tamil's complex agglutinative morphology with extensive case marking and verb conjugations genuinely requires the higher representational capacity of mT5-base (580M parameters, 12 layers) to capture and correct diverse morphological error patterns effectively. The smaller model struggled with complex morphological transformations, producing more errors in handling multi-token corrections and verb form agreements.

Conversely, for Malayalam, increasing model capacity from mT5-small to mT5-base yielded a marginal performance difference (55.21% vs. 55.03%, a negligible -0.18 delta), confirming that the larger model provides no substantial benefit for Malayalam GEC in our constrained data setting. Critically, preliminary analysis revealed that mT5-base for Malayalam exhibited increased generation instability, producing more instances requiring safety validation rejection compared to mT5-small. This behavior validates our conservative approach: mT5-small's lower capacity, combined with strict safety mechanisms (token overlap $\geq$45%, length ratio $\geq$50%, deletion ratio $\leq$45%), provides an optimal balance between correction capability and output reliability for Malayalam.

These findings demonstrate that our model selection was empirically grounded rather than arbitrary: Tamil benefits substantially from higher model capacity to handle morphological complexity, while Malayalam requires conservative capacity with robust safety validation to prevent overcorrection in extremely low-resource settings. The asymmetric capacity requirements reflect fundamental differences in how the two languages manifest errors and respond to neural correction in data-scarce scenarios.

5 Discussion

Extremely low resource GEC requires hybrid approaches with optimal balance between the neural and symbolic rule-based components depending on the language characteristics, dataset size, and deployment priorities. The key advantage of the hybrid architecture lies in its application of a selective strategy with high-precision rules, handling known patterns with perfect accuracy, neural generation providing fallback for unseen error types, and post-processing refining neural outputs while preventing common failures.

Tamil system success stemmed from mT5-base providing sufficient capacity for complex Tamil morphology without extreme overfitting, conservative augmentation to 5,000 examples preventing noise pattern learning, manual pattern curation compensating for training data gaps, and hierarchical correction strategy ensuring deterministic handling of known patterns, achieving 85.34% GLEU on test data. Malayalam system success resulted from mT5-small with strict safety validation preventing overfitting and output degeneration, aggressive but filtered augmentation to 10,000 examples maintaining quality, multi-layered safety framework ensuring output quality preservation, and parallel processing with conservative fallback prioritizing reliability, achieving 95.06% GLEU and securing Rank 2.

While direct comparison remains limited due to novel datasets, our results surpass reported low-resource GEC performance. Czech GEC with synthetic augmentation achieved approximately

Table 2: Error Analysis – Tamil and Malayalam

Input Sentence	Hybrid Output	Correction Type
தொழிற்சாலை இயந்தித்தின் சத்தம் thozhilsaalai iyandhithithin sattham / ”factory mashine’s noise”	**தொழிற்சாலை இயந்திரத்தின் சத்தம்** thozhilsaalai iyanthirathin sattham / ”factory machine’s noise”	**Morphological** இயந்தித் → இயந்திர (iyandhithith → iyanthira)
போக்குவரத்து வாகணங்களின் ஹாரன் pokku varatthu vaakanangalin haaran / ”traffic vehikles’ hron”	**போக்குவரத்து வாகனங்களின் ஹார்ன்** pokku varatthu vaahanangalin haarn / ”traffic vehicles’ horn”	**Multi-token Correction** haaran → haarn; வாகணம் → வாகனம் (vaakanam → vaahanam)
நமது காது பாதிப்படைகின்றன namathu kaathu paathippadaikindrana / ”our ear gets affected” (plural verb)	**நமது காது பாதிக்கப்படுகிறது** namathu kaathu paathikkappadugiradu / ”our ear gets affected” (singular verb)	**Verb Form + Subject-Verb Agreement**
கல்லுாரியையும் கல்லுாரி நாட்களைப் பற்றியும் kalluurariyaiyum kalluurari naalkallaip patriyum / ”colege and colege days about”	**கல்லூரியையும் கல்லூரி நாட்களைப் பற்றியும்** kallūriyaiyum kallūri naalkallaip patriyum / ”college and college days about”	**Vowel Length Normalization** ூ → ூ uaa → -ū
ஒலிப்பெருக்கியை இரவு நேரத்தில் பயன்படுத்தாமல் அவசியம் இருப்பது olipperukkiyai iravu neraththil payanpaduththaamal avasiyam iruppadhu / ”loudspeaker night time not using necessary is”	**ஒலிப்பெருக்கியை இரவு நேரத்தில் பயன்படுத்தாமல் இருப்பது அவசியம்** olipperukkiyai iravu neraththil payanpaduththaamal iruppadhu avasiyam / ”loudspeaker night time not using is necessary”	**Word Order Reordering** + Marker Preserved
സമുദ്രത്തിൽ തള്ളുന്ന കഴിവുകൾ samudraththil thaḷḷunna kazhivukaḷ / ”ocean dumping wastes”	**സമുദ്രത്തിൽ തള്ളുന്ന കഴിവുകൾ** samudraththil thaḷḷunna kazhivukaḷ / ”ocean dumping wastes”	**Preserved** (no correction needed)
വാകണം ഓടിച്ചു vaakanam odichchu / ”vehikle drove”	**വാഹനം ഓടിച്ചു** vaahanam odichchu / ”vehicle drove”	**Spelling correction** വാകണം → വാഹനം/ vaakanam → vaahanam
കടലിൽ മലിനീകരണം കാരണം kadalil malineekaranam kāranam / ”sea pollution reason”	**കടലിൽ മലിനീകരണം കാരണം** kadalil malineekaranam kāranam / ”sea pollution reason”	**Preserved** with validation
ധ്വനി മലിനീകരണത്തിന് കാരണങ്ങൾ dhvani malineekaranaththinu kāranaṅṅaḷ / ”noise pollution’s reasons”	**ധ്വനി മലിനീകരണത്തിന് കാരണങ്ങൾ** dhvani malineekaranaththinu kāranaṅṅaḷ / ”noise pollution’s reasons”	**Token-level preservation**

60-70% accuracy with similar data constraints (Naplava & Straka, 2019), while our hybrid approach achieved 85.34% GLEU for Tamil and 95.06% GLEU for Malayalam, demonstrating viability for extreme low-resource scenarios. Key advantages include high-precision rules handling known patterns with perfect accuracy, neural generation providing fallback for unseen error types, post-processing refining neural outputs and preventing common failures, and conservative safety mechanisms ensuring real-world deployability.

6 Conclusion

We successfully presented a robust unified neurosymbolic framework for Grammatical Error Correction in extremely low-resource Indic languages, applying it to Tamil and Malayalam. By strategically differentiating neural model capacity and ensemble strategy, we optimized for unique challenges of each language. These systems prove that combining modern pre-trained models, parameter-efficient fine-tuning, aggressive augmentation, and linguistic rule engineering provides a powerful practical approach for GEC when facing severe constraints on annotated data, serving as a blueprint for developing GEC systems for low or under-resourced Indic languages.

7 Limitations and Future Work

Limitations include small training and validation datasets. This limits statistical confidence, pattern coverage that cannot address all possible grammatical errors, especially rare or domain-specific mistakes, risk of pattern memorization rather than

generalizable learning, and system assumptions regarding formatting conventions or safety thresholds that may not cover all use cases.

Future directions should focus on larger evaluation datasets enabling statistically reliable performance assessment, cross-domain testing on different text types, linguistic integration incorporating explicit morphological and syntactic knowledge, active learning to identify high-value training examples, cross-lingual transfer leveraging knowledge between related Dravidian languages, automated pattern discovery reducing reliance on manual curation, and adaptive mechanisms enabling dynamic threshold adjustment.

8 Acknowledgement

We thank the IndicGEC shared-task organizers and annotators for providing data and evaluation tools. Our gratitude extends to the Tamil and Malayalam NLP communities for sustained linguistic resource development and to open-source frameworks enabling this research.

References

[1] Awasthi, A., Sarawagi, S., Goyal, R., Ghosh, S., & Piratla, V. (2019). Parallel iterative edit models for local sequence transduction. In Proceedings of EMNLP-IJCNLP, (pp. 4260-4270).

[2] Bryant, C., Felix, M., Andersen, E., & Briscoe, T. (2019). The BEA-2019 shared task on grammatical error correction. Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications, (pp. 52-75).

[3] Flachs, S., Hardt, D., & Sogaard, A. (2021). Assessing text readability for second language learners. In Proceedings of the 16h Workshop on Innovative Use of NLP for Building Educational Applications, (pp. 95-100).

[4] Grundkiewicz, R., Junczyz-Dowmunt, M., & Heafield, K. (2019). Neural grammatical error correction systems with unsupervised pre-training on synthetic data. Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications, (pp. 252-263).

[5] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., . . . Chen, W. (2021). LoRA: Low-rank adaptation of large language models. In Proceedings of ICLR.

[6] Joshi, N., Darbari, H., & Mathur, I. (2012). HMM based POS tagger for Hindi. In Proceedings of the 2012 International Conference on Artificial Intelligence.

[7] Kaneko, M., Mita, M., Kiyono, S., Suzuki, J., & Inui, K. (2020). Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In Proceedings of ACL, (pp. 4248-4254).

[8] Katsumata, S., & Komachi, M. (2020). Stronger baselines for grammatical error correction using a pretrained encoder-decoder model. In Proceedings of AACL-IJCNLP, (pp. 827-832).

[9] Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvinineiad, M., . . . Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. Transactions of the ACL, 726-742.

[10] Mitra, A., & Baral, C. (2016). Learning to use formulas to solve simple arithmetic problems. In Proceedings of ACL, (pp. 2144-2153).

[11] Naplava, J., & Straka, M. (2019). Grammatical error correction in low-resource scenarios. In Proceedings of the 5th Workshop on Noisy User-generated Text, (pp. 346-356).

[12] Platanios, E. A., Pauls, A., Roy, S., Tsvetkov, Y., & Klein, D. (2021). Competence-based curriculum learning for neural machine translation. In Proceedings of NAACL-HLT, (pp. 1018-1032).

[13] Rothe, S., Mallinson, J., Malmi, E., Krause, S., & Severyn, A. (2021). A simple recipe for multilingual grammatical error correction. In Proceedings of ACL-IJCNLP, (pp. 702-707).

[14] Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., . . . Raffel, C. (2021). A massively multilingual pre-trained text-to-text transformer. In Proceedings of NAACL-HLT, (pp. 483-498).

[15] Zhao, W., Wang, L., Shen, K., Jia, R., & Liu, J. (2019). Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In Proceedings of NAACL-HLT, (pp. 156-165).

[16] Rajalakshmi, R., Sharma, V., M, A.K. (2023). Context Sensitive Tamil Language Spellchecker Using RoBERTa. In: M, A.K., et al. Speech and Language Technologies for Low-Resource Languages. SPELLL 2022. Communications in Computer and Information Science, vol 1802. Springer, Cham. https://doi.org/10.1007/978-3-031-33231-9_4