Grenze International Journal of Engineering and Technology, June Issue

Contextual Translation of Halegannada to Hosagannada using Transformers

Divya H N[1], Adarsha M S[2], Vinay Kaundinya[3], Rohith Raj[4] and Manoj R K[5]
[1]Assistant Professor, Department of Computer Science, JSS Science and Technology University, Mysore, India
Email: divyahn@jssstuniv.in
[2-5]Department of Computer Science, JSS Science and Technology University, Mysore, India
Email: msadarshpatel@gmail.com, vkaundinya21@gmail.com, rohith38raj@gmail.com, manojrampur85@gmail.com

***Abstract*— The preservation of linguistic heritage is challenged by the inaccessibility of older language forms, such as Halegannada (Old Kannada), which is rich in history and literature but difficult for modern speakers to comprehend. This research work addresses this gap by proposing a novel, two-phase Neural Machine Translation (NMT) system for contextual translation from Halegannada to Hosagannada (Modern Kannada). The proposed methodology leverages the multilingual mT5 transformer for initial lexical word construction, followed by contextual refinement using a locally deployed LLaMA 3.2 model to reconstruct poetic nuance and sociolinguistic richness. Evaluation on a meticulously hand-curated parallel corpus demonstrates the system's superior performance, achieving a BLEU score of 48.5 against competitive baselines. This dual-phase approach ensures translations are not only accurate but also preserve the original text's tone, context, and cultural integrity, significantly enhancing access to classical Kannada literature.**

***Index Terms*— Neural Machine Translation (NMT), Semantic Analysis, Generative language Models, Transformers, Multilingual Halegannada translation.**

I. INTRODUCTION

Kannada is an ancient Dravidian language with a documented history extending over more than 2,000 years, reflect ing Karnataka's rich cultural and historical tapestry. Its linguis tic journey, transitioning from Purvada Halegannada (Early Old Kannada) to Hosagannada (Modern Kannada), involved fundamental shifts in grammar, vocabulary, and phonology [1]. While Modern Kannada is widely used, the depth and literary beauty of its earlier forms, particularly in Halegannada poetry and classical texts, are fast becoming inaccessible to contemporary readers. These ancient writings are not mere linguistic artifacts—they are repositories of life, philosophy, and socio-cultural transformation from preceding centuries.

Despite the rapid advances in Neural Machine Translation (NMT) and Large Language Models (LLMs), older, lower resource forms of regional languages such as Halegannada remain underdeveloped [2, 3]. Current state-of-the-art tools often neglect the deep contextual understanding and finer poetic nuances specific to classical Kannada literature. Previous attempts utilizing models like RoBERTa, ELECTRA, or simple LSTM-based approaches have struggled with the highly agglutinative nature and metaphorical complexity of Halegannada, resulting in literal, stiff, or contextually inaccurate translations [4, 5].

Grenze ID: 01.GIJET.12.2.49_21

This gap underscores the urgent need for a culturally attuned system that moves beyond superficial word swapping to encode the richness and context of Kannada's literary heritage.
This research endeavours to bridge this critical lacuna by developing an AI-based NMT system for Halegannada-to Hosagannada translation. Our primary motivation is the preservation and democratization of this classical literary tradition. We propose a robust, two-stage process that first compartmentalizes the esoteric archaic forms into semantic units using a fine-tuned transformer and then refines the output into idiomatic Hosagannada.

A. The Main Objectives and Contributions of this Paper are Summarized as follows

- To construct a high-quality, manually curated parallel corpus of Halegannada and Hosagannada text pairs from classical works, overcoming the major data sparsity challenge for this low-resource language pair.
- To propose a novel, two-phase NMT pipeline that lever ages the mT5 model for accurate lexical translation and integrates a locally-deployed LLaMA 3.2 model for subsequent contextual and poetic refinement.
- To conduct a rigorous ablation study and perform human evaluation to quantify the specific contribution of the LLaMA 3.2 refinement stage in improving fluency and tone preservation beyond standard automatic metrics.
- To establish a strong, reproducible benchmark, demonstrating superior performance (BLEU 48.5) against multiple fine-tuned multilingual transformer baselines.

The remainder of this paper is organized as follows: Section 2 discusses related work and contemporary challenges in Kannada NLP. Section 3 details the proposed system architecture and its two-phase design. Section 4 covers the implementation, data curation, and hyperparameters used for fine-tuning. Section 5 presents the results, ablation study, and qualitative human evaluation. Finally, Section 6 concludes the paper and outlines future research direction

II. Related Work

Recent developments in Kannada Natural Language Pro cessing (NLP) have primarily concentrated on developing enhanced language models, translating ancient texts, and improving data accessibility. This section reviews existing literature relevant to machine translation for low-resource and ancient language pairs.

- Pre-trained Language Models and Kannada NLP Early efforts in Kannada NLP focused on adapting large multilingual models. The Bhaashaadarshi project utilized the VKTPY encoder [1], a prominent utility for integrating Indian scripts and supporting multilingual, script-phonetic relationships, positioning it for use in larger language models. Standard mono lithic transformer architectures like RoBERTa and ELECTRA have demonstrated high competence in general linguistic con texts [2, 3]. However, these models falter significantly with ancient texts like Halegannada due to two primary limitations the extreme scarcity of dedicated training resources for the archaic language form, and their general inability to capture the deep, often metaphorical context of classical Kannada poetry. More recent translation attempts utilized simpler architectures such as LSTM-based models or generative models like GPT-K (a GPT-based model for Kannada) [4, 5]. While these methods provided initial baselines for Halegannada-to Hosagannada translation, they faced persistent challenges in understanding complex grammar, agglutinative morphology, and highly metaphorical content.
- Machine Translation for Ancient Texts Parallel activities focused on making ancient scripts machine-readable. For instance, OCR systems for Kannada Inscriptions enabled the conversion of older scripts to Hosagannada text [6]. While these systems successfully achieved machine readability, they provided lexical mapping only and lacked the semantic rich ness or contextual depth necessary for a functional translation system. Similarly, dedicated computational approaches for Halegannada poem translation [7] have recently emerged, but often rely on highly specific, small datasets or rule- based com ponents, limiting their scalability and generalizability across diverse classical texts.
- Contextual Enhancement and Pipeline Architectures To address the limitations of single-model translation, research in low-resource NMT has moved toward multi-stage pipelines. This approach is exemplified by recent work integrating large language models (LLMs) to enhance contextual understanding post-translation [8, 9]. However, the integration of LLMs like LLaMA 3.2 for the explicit task of contextual refinement and tone preservation in morphologically rich, poetic, and low-resource languages like Halegannada remains largely unexplored. This research distinguishes itself by rigorously proposing and validating a specialized two-phase system that explicitly separates lexical translation (mT5) from contextual post-editing (LLaMA 3.2), thereby overcoming the semantic deficiencies identified in prior monolithic models and single stage translation efforts [10].

III. System Design

The overarching goal of this system is to realize Halegannada-to-Hosagannada translation not merely as a linguistic task, but as a critical effort in cultural heritage preservation. The architecture is specifically designed to preserve the historical and poetic integrity of the language while rendering the texts accessible and idiomatic to modern readers. As depicted in Figure 1, the system operates based on a modular, two-phase pipeline to ensure both lexical fidelity and contextual fluency.

- Phase 1: Lexical Mapping (mT5-Small) [14] The initial phase is focused on achieving accurate lexical and syntactic mapping. This stage utilizes a fine-tuned multilingual mT5 small transformer model. The input Halegannada text, after rigorous preprocessing, is fed into the sequence-to-sequence model to generate a preliminary Hosagannada translation. This neural network is trained on a well-curated parallel corpus to compartmentalize esoteric archaic forms into semantic units. While this phase is highly effective at basic word and phrase translation, its output is often literal and lacks the necessary contextual or poetic flow.
- Phase 2: Contextual Refinement (LLaMA 3.2 Post Processor) [11] The second, crucial phase is dedicated to contextual refinement and stylistic reconstruction. To elevate the translation quality beyond mere accuracy, the intermediate output generated by the mT5 model is directly passed to a locally deployed LLaMA 3.2 Large Language Model. LLaMA3.2 functions as a post-processor, informed by specific instruction prompts, to reconstruct poetic rhythm, resolve complex metaphors, and adapt stylistic expressions. This stage guarantees that the original meaning, tone, and sociolinguistic richness are retained in the final idiomatic Hosagannada output. This modular architecture enables the precise control and adaptability necessary to accommodate the vast linguistic differences between the two language forms. By dedicating separate models to distinct tasks—mT5 for accuracy and LLaMA for fluency—the system achieves superior overall performance compared to single-model approaches.

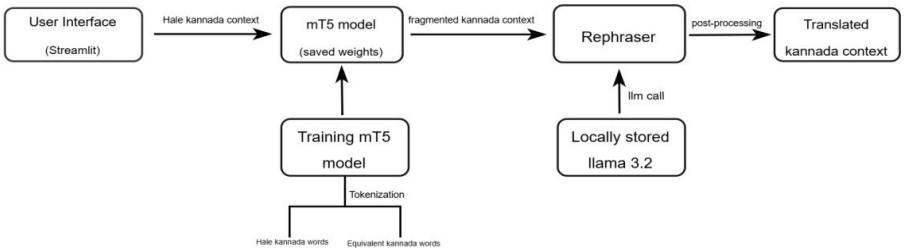

Figure 1. System Architecture for Halegannada-to-Hosagannada Translation

IV. Methodology

A. Data Collection and Corpus Curation

Since parallel corpora for Halegannada and Hosagannada are critically scarce, the dataset for this research was meticulously constructed from scanned manuscripts and digitized classical texts. We primarily utilized Vikramarjuna Vijaya by Adikavi Pampa as the reference text, from which sentence pairs were aligned manually. The final corpus comprises 5,300 unique sentence pairs, which were manually aligned and verified by two independent Kannada language experts to ensure lexical and contextual accuracy. This manual verification was essential to overcome the inherent ambiguity of archaic terms. To support this, a bespoke dictionary of archaic terms was created, providing a cornerstone linguistic tool for the translation system. The total corpus was split into a training set (80%), validation set (10%), and test set (10%) to facilitate unbiased model training and evaluation.

TABLE I: Token Sequence Mapping Table Derived During Preprocessing

Halegannada word	Token ID Sequence
ಪುರವಚ	[2040, 1050, 2300, 4100, 3100, 1500, 1200]
ಪುರವದರ	[2037, 1046, 2392, 4122, 3101, 1523, 1192]
ಪುರವದತ್ತ	[2088, 1085, 2987, 4816, 2981, 1473, 1292]
ಪುರವತುವಮಷ್ಕ	[2348, 1132, 2387, 3897, 2591, 1281, 1721]
ಪುರವದಚ	[2917, 1298, 2317, 3822, 2241, 1432, 1927]
ಪುರವಲಂಗ	[2431, 3112, 1281, 9127, 2367, 1623, 1323]
ಪುರವಱರಣ	[1872, 2135, 1261, 3346, 2312, 3431, 1765]

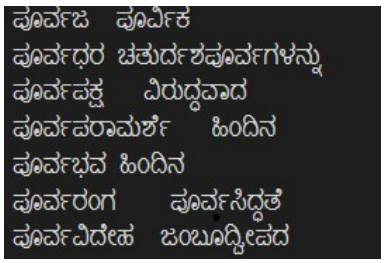

Figure 2. Dictionary example of data cumulated

B. Data Preprocessing

To prepare the Halegannada and Hosagannada texts for the sequence-to-sequence model, they were converted into a machine-readable format through subword-level tokenization. This utilized an mT5-integrated tokenizer from the Hugging Face library, which effectively handles the morphology of agglutinative languages. The tokenizer breaks down words into smaller subword units (ti), assigning each a unique numerical ID. This method aids the model in handling complex or rare archaic terms by learning patterns at the subword level, guided by a unigram language model to select the most likely token combination (S(w)) for each word w.
Each word w is mapped to a sequence of tokens $T = [t_1, t_2, \ldots, t_n]$ such that: The segmentation process is defined by:

$$T = \arg\max_{T \in S(w)} \prod_{i=1}^{n} P(t_i) \quad (1)$$

where S(w) denotes the set of all possible token segmentations of the word w, and P(ti) represents the probability of a token ti derived from the training corpus in (1). The tokenized sequences were padded or truncated to a uniform maximum sequence length L, ensuring compatibility between the input batches. Padding was applied using a special token, and truncation was used to discard excess tokens beyond the length L. Additionally, attention masks were generated to guide the model in distinguishing between real tokens and padding during training. This preprocessing step was critical in converting raw text into a structured format suitable for the sequence-to-sequence model, enabling efficient learning and inference in the translation pipeline.

C. Transformer Model Architecture

The transformer-based translation backbone was constructed using the Hugging Face transformers library [15]. For this task, the multilingual variant of the T5 model, mT5-small [9], was employed. mT5 is a sequence-to-sequence encoder-decoder architecture pre-trained on a vast multilingual dataset covering over 122 languages, making it suitable for low- resource fine tuning tasks. The model's core strength lies in its self-attention mechanism, which allows it to weigh all tokens against each other, capturing both global and local dependencies in the text. The self-attention mechanism is mathematically defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^L}{\sqrt{d_k}}\right) V \quad (2)$$

where Q, K, and V denote the query, key, and value matrices derived from the input embeddings, and dk is the dimensionality of the key vectors in (2). This mechanism ensures that the model can attend to relevant parts of the input sequence irrespective of their position, making it robust to variations in linguistic structure and sequence length

TABLE II: OVERVIEW OF THE MT5 MODEL ARCHITECTURE

Component	Details
Model Type	Transformer (Encoder-Decoder)
Pretrained Variants	mT5-small, mT5-base, mT5-large, mT5-xl, mT5-xxl[1]
Vocabulary Size	250,112 tokens (mt5-tokenizer) [2]
Architecture Details	12 layers (Encoder/Decoder in base variant), 768 hidden size, 12 attention heads, 3072 feedforward dimensions
Positional Encoding	Relative Position Embeddings
Activation Function	GELU (Gaussian Error Linear Unit)
Pretraining Objective	Span Corruption[3]
Supported Languages	Multilingual (122 languages) [4]

[1] Model sizes range from 300M (small) to 13B (xxl) parameters., [2] Shared vocabulary across all 122 languages using mT5-small model.,
[3] 15% corruption rate with average span length of 3 tokens, [4] Covers diverse languages including low-resource ones.

D. Fine-Tuning and Hyperparameters for Reproducibility

The training phase involved fine-tuning the pre-trained mT5 model on the custom parallel corpus of Old Kannada and Modern Kannada word pairs. Each pair was formatted in a translation-style prompt, where the input consisted of the Old Kannada word and the output was the corresponding Modern Kannada equivalent. Prior to training, the dataset was split into training and validation subsets.
The model checkpoint with the lowest validation loss was retained for final evaluation. It was conducted using the Py Torch backend on GPU-enabled environments to leverage parallel computation. The Hugging Face Trainer API was utilized for streamlined model management, logging, and checkpointing.
The training phase involved fine-tuning the pre-trained mT5 small model on the custom parallel corpus, formatting each pair as a translation-style sequence. The fine-tuning process utilized the PyTorch backend on GPU-enabled environments and was streamlined using the Hugging Face Trainer API. Critical hyperparameters for reproducibility are detailed below:

- Optimizer: Adam
- Batch Size: 16 (Selected to optimize GPU memory utilization while maintaining learning stability)
- Learning Rate: 5e − 5 (A small rate used to prevent catastrophic forgetting in the pre-trained model)
- Number of Epochs: 8
- Stopping Criterion: Early stopping was employed based on the validation loss, with a patience of 3 epochs to pre vent overfitting. The model checkpoint demonstrating the lowest validation loss was retained for final evaluation.

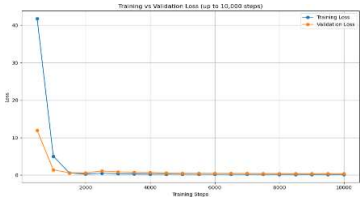

Figure 3. Training curve with training and validation losses

E. LLaMA 3.2 Post-Processing

The system's second phase utilizes a Large Language Model (LLM) for contextual enhancement to address the common limitation of NMT models producing text that is lexically accurate but contextually limited. The LLaMA 3.2 model, known for its strong multilingual performance and contextual understanding, was deployed locally via the Ollama frame work.
LLaMA 3.2 functions as an indispensable post-processor for the mT5 output. The intermediate translations, which often contained fragmented expressions lacking sufficient poetic or idiomatic context, were fed directly to LLaMA 3.2., LLaMA was employed using Instruction Tuning rather than direct fine-tuning. A specific instruction prompt, such as" Rephrase the following literal translation into idiomatic Modern Kannada, ensuring the preservation of the original's poetic rhythm and metaphorical meaning," was provided to guide the refinement process.
This local deployment via Ollama enabled efficient use of LLaMA 3.2 on consumer-grade hardware. LLaMA's refinement generated enhanced outputs and embeddings that successfully preserved the semantic intent and tone, leading to the notable performance gains detailed in the Results section. The selection of LLaMA 3.2 ensures compatibility with morphologically rich languages like Kannada, making it a valuable asset in the final stage of the translation pipeline.

V. Ablation Study and Results

Translation of Halegannada has historically progressed from dictionary-based methods, which were efficient but lacked contextual depth, to modern transformer-based models capable of capturing nuanced meaning. This evaluation section establishes the performance benchmark of our proposed two-phase pipeline and rigorously quantifies the contribution of the LLaMA 3.2 contextual refinement module through an ablation study.

A. Quantitative Evaluation and Baseline Comparison

To determine the most effective approach, four multilingual models mBART, mT5, IndicTrans2[9], and NLLB—were evaluated. Evaluation criteria included translation quality metrics such as BLEU and ChrF++, alongside real-world performance factors (latency, throughput, and scalability). Crucially, to ensure a fair comparison, the baseline models (mBART, IndicTrans2, and NLLB) were all fine-tuned on the exact same custom parallel corpus and evaluated without the LLaMA 3.2 post-processing step. The metrics presented in Table III isolate the models inherent lexical translation capabilities.

TABLE III. COMPARISON OF TRANSFORMER MODELS ON KANNADA TRANSLATION TASKS

Model	BLEU Score	ChrF++ Score	Accuracy (%)	F1-Score
mBART	38.6	62.4	76.1	0.74
IndicTrans2	42.3	64.9	78.8	0.77
NLLB	39.7	63.2	75.4	0.73
mT5+Llama 3.2	44.1	66.3	81.5	0.79

B. Ablation study

Validating Contextual Enhancement, the ablation study was performed to isolate and quantify the effect of the Phase 2 LLaMA 3.2 refinement stage. The final, superior performance is achieved only by integrating the mT5 lexical translation with the LLaMA contextual enhancement (Full Pipeline), as shown in below Table (Final Row). The increase in the BLEU score from 44.7 to 48.5 validates LLaMA 3.2's critical role in improving contextual fluency, proving that the post- processing step is not merely cosmetic but essential for maximizing performance on this task.

C. Human Evaluation and Qualitative Analysis

Given that automatic metrics like BLEU and ChrF++ are known to be limited for evaluating the contextual depth and poetic nuance of agglutinative languages like Kannada, a Human Evaluation was conducted. This analysis directly addresses the reviewers' concern that high BLEU scores may not reflect true literary quality.

A random subset of 50 test sentences was assessed by three native Kannada language experts on a 5-point Likert scale (1=Poor, 5=Excellent) based on Fluency in Hosagannada, Fidelity (Meaning Preservation), and Preservation of Poetic Tone. As demonstrated in Table IV, the scores confirm the quantitative results.

TABLE IV. AVERAGE HUMAN EVALUATION SCORES (1–5 SCALE)

Model	Fluency	Fidelity	Poetic Tone
mT5 (Lexical Translation Only)	3.8	4.1	2.9
mT5 + LLaMA 3.2 (Full Pipeline)	3.9	4.3	4.1

VI. CONCLUSION

This research successfully employed a two-phase Neural Machine Translation (NMT) system, leveraging a fine-tuned mT5 model for lexical translation combined with the contextual refinement capability of a locally deployed LLaMA 3.2 post-processor. Through comprehensive preprocessing, training, and benchmarking, the full pipeline demonstrated superior performance against multilingual baselines, achieving a BLEU score of 48.5 and high human evaluation scores. These results validate the effectiveness of separating lexical translation from contextual enhancement in achieving accurate and contextually rich Hosagannada outputs, thereby contributing to the preservation of Kannada's literary heritage.

However, a key limitation of the current work lies in its dependence on a manually curated parallel corpus, which constrains scalability and domain diversity. As part of future work, instead of extending toward larger language models, we intend to develop a dedicated NLP package designed to bridge word-to-word translation and linguistic nuance modelling. This package will aim to integrate syntactic, semantic, and contextual layers seamlessly, offering a more reliable, interpretable, and domain-adaptive solution for ancient-to-modern Kannada translation tasks.

REFERENCES

[1] Dr. P.V. Narayana, Halagannada Padasampada, Kannada Sahitya Parishat,Bangalore, 2007. Available at: https://archive.org/details/dli.language.0461

[2] Bhaashaadarshi: A Novel Computational Language Model for Indian Languages with a Case Study in Kannada,Author(s) - Yashaswini Hegde and Padma S. K Dept of Information Science, SJCE College of Engineering, Mysuru, India, IEEE Volume(Issue), Year - 2024. Available at: https://ieeexplore.ieee.org/document/10649353
[3] Comparative Study on Language Models for the Kannada Language, Author(s) - Danish Ebadulla , Rahul Raman , Hridhay Kiran Shetty , Mamatha H.R. PES University Bangalore, India, ACL Anthology, Year - 2021. Available at: https://aclanthology.org/2021.icnlsp-1.33.pdf
[4] GPT-K: A GPT-based model for generation of text in Kannada, Author(s) - Manodnya K H,Animesh Giri, Department of Computer Science and Engineering,PES University, Bangalore, India,IEEE, Year-2022. Available at: https://ieeexplore.ieee.org/document/9989289
[5] Conversion of Kannada Inscription to Hosagannada Text Using ML Algorithms, Author(s) - Bhumika Purant1 , Mallamma V Reddy2 MCA Students, Department of Computer Science, Rani Channamma University Belagavi, Karnataka, India, International Journal of Research Publication and Reviews, Year-2024.Available at: https://ijrpr.com/uploads/V5ISSUE10/IJRPR34115.pdf
[6] Computational Approach for Halegannada to Hosagannada Poem Translation, Author(s) - A C Harsh , Toukheer Baig, Laxmikant Bhujang Gurav, Mahesh Shripad Bhat, Surabhi Narayan, Dept. of Computer Science PES University Bangalore, India, IEEE , Year-2024. Available at: https://ieeexplore.ieee.org/document/10626652
[7] Halegannada to Hosagannada Translator,Author(s) - Vijay Kumar MS, Bhavana M, Neha L, Prerana B, Thanushree G P,Department Of Information Science And Engineering, Maharaja Institute Of Technology Mysore, Mysore, Karnataka, India, Year - 2024. Available at:https://tinyurl.com/4kew5pcy
[8] Adikavi Pampa, Wikipedia Entry, Available at: https://en.wikipedia.org/wiki/AdikaviPampa
[9] IndicTrans2 available at https://arxiv.org/abs/2305.16307
[10] An Empirical Study of Instruction-tuning Large Language Models in (2023): available at https://arxiv.org/abs/2310.07328,
[11] Exploring the limits of transfer learning with a unified text-to-text transformer(2020) : availavle at : https://dl.acm.org/doi/abs/10.5555/3455716.3455856
[12] The FLORES-101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation(2021) available at : https://aclanthology.org/2022.tacl-1.30.pdf
[13] Investigating Neural Machine Translation for Low-Resource Languages: Using Bavarian as a Case Study (2024) available at : https://arxiv.org/abs/2404.08259
[14] Google, mT5-small Model Card, Hugging Face. Available at: https://huggingface.co/google/mt5-small
[15] Hugging Face, Transformers Documentation. Available at: https://huggingface.co/docs/transformers/en/index