# **ACCESSIBLE TAMIL SANGAM LITERATURE USING ARTIFICIAL INTELLIGENCE**

# **INTRODUCTION**

Tamil is one of the longest surviving classical languages spoken by Tamil people in South Asia and Tamil diaspora across the world. [ADD REFERENCE FROM WIKI]

Tamil Sangam literature (also known as the poetry of the noble ones) is the earliest known literature of South India. The Sangam literature era spanned from c.300 BCE to 300CE. Tamil Sangam literature is broadly classified into Akam (அகம், inner) and Puram (புறம், outer). Agam literature focuses on feelings/emotions of romantic love, ethics, morality, nature and similar ones. Puram literature focuses on heroic deeds of war and public life. These literatures also serve as historical documents for many notable events and people, with vivid description.

One of the literatures that is most significant of all is Thirukural (திருக்குறள்). Thirukural contains 1330 short couplets (4 words in the first line and 3 words in the second line) which focuses on moral and ethics, and it is one of most translated ancient literature works.

In short, Tamil literature is a treasure trove of poems that reflects vibrant Tamil culture, inclusivity of both men and women in Tamil society, ethical and moral values upheld by them and anecdotes of historical events and people.

Anyone who understand these literatures can learn many virtues and improve their life.

However, Tamil we use today has changed significantly over time and it is not possible to understand/enjoy Tamil Sangam literature by anyone but Tamil scholars. This is mainly due to the changes to vocabulary, most words used in Tamil Sangam literature is not used in modern Tamil and things like metaphors, similes, and imageries used in the Tamil Sangam literature for poetic elegance requires historical and contextual knowledge to understand and uncover the true meanings.

There are LLMs that can take prompts in modern Tamil and respond in modern Tamil, upon evaluation, they are significantly inadequate when using them to understand/translate Tamil Sangam literature in modern Tamil.

The aim of this study is to explore the possibility of making Tamil Sangam literature accessible to anyone who can read and understand modern Tamil using cost-effective fine-tuning techniques and evaluate their performances.

Currently, the interest in Tamil literature work is fading, mainly because English dominates the internet and almost all technological advancements. This research would bring latest technological advancements to Tamil too, thereby, reviving interest in Tamil literature by making it accessible to modern Tamil people.

The outcome of this study is fine-tuned Tamil LLMs that takes Tamil Sangam literature work as prompt and output the explanation in modern Tamil at inference time. The models are made publicly available to enable further exploration in this field.

Also, the datasets curated for this work are made available for anyone to use for their own research and experimentation.

The quality of the outcome is verified by combining two different approaches - automatic evaluation using ROUGE (Lin, 2004), BLEU (Papineni et al., 2002) CHRF and Meteor metrics and Human verification to ensure quality, coherence and simplicity.

This research work is built on top of Tamil LLaMA, an excellent work by Balachandran, 2023, and Tamil Mistral by Hemanth Kumar, 2024.

# **RELATED WORK**

In this section, we cover the details of existing Tamil LLMs and other low resource language modes that formed the foundation for this study.

## **2.1 TAMIL LLAMA**

Tamil LLaMA model was created by enhancing LLaMA 2 7B model with additional 16,000 Tamil tokens. These tokens are made up of root words, lemma and suffixes. The model was then fine-tuned using LoRA for computational and cost efficiency and SentencePiece was used for tokenization.

Tamil translated versions of Alpaca dataset and OpenOrca datasets are used for fine-tuning LLaMA 2.

It was found that the LLaMA 2 7B model enhanced with additional Tamil tokens showed improved performance in understanding and generating Tamil text. The use of LoRA for model training proved to be effective in fine-tuning the model with limited computational resources. The translated Alpaca dataset and OpenOrca data significantly contributed to the model’s ability to follow instructions and generate coherent responses in Tamil. The fine-tuned model demonstrated capabilities comparable to proprietary models like GPT-4 in specific tasks like summarisation. This work emphasised the potential of smaller, fine-tuned models to achieve high performance with targeted training data. This finding is crucial for the current study as we are using similar targeted training data technique for Tamil Sangam literature.

The outcomes are evaluated using GPT-4, which assigned grade on a 10-point scale. In addition to this Balachandran, 2023 also employed manual human verification to validate the quality of the outcome.

Although the model was enriched by 16,000 new Tamil token, this is not adequate for all tasks – especially for translating classical Tamil. This is for two reasons - first the 16,000 additional tokens only include most common tokens from modern Tamil, but classical Tamil uses many tokens that do not exist in modern Tamil, second the classical Tamil poems words are deliberately split at seemingly arbitrary places for meter, rhythm and aesthetic structure of the poem.

The dataset used for fine-tuning covers various domains, but Tamil Sangam literature was not part of the dataset.

As a result, the model is inadequate when used for explaining Tamil Sangam literature and not performing very well when it comes to understanding complex linguistic nuances and idiomatic expressions in Tamil.

## **2.2 TAMIL MISTRAL**

Tamil Mistral (Hemanth Kumar, 2025) was created by fine-tuning Mistral-7B-Instruct (Jiang et al., 2023) modal using 400k Tamil instruction dataset.

The performance and findings of this model was not published but during experimentation, it was found that this model performs well in taking modern Tamil input as prompt and generating modern Tamil at inference time. However, this model also suffers from same challenges we face with Tamil LLaMA, the model is not adequate to when it comes to translating Tamil Sangam literature.

However, we now know from the work by both Balachandran, 2023 and Hemanth Kumar, 2025 that pre-trained LLMs can successfully be fine-tuned for Tamil language. This gives us a strong foundation and high confidence that it is worth exploring fine-tuning these models on Tamil Sangam literature instruction dataset.

## **2.3 OTHER LOW RESOURCE LANGUAGES WORKS**

Tamil is one of the low resource languages, although Tamil has vast collection of data, they are not in digital medium which means curating good quality dataset is a challenge.

[sphinx] highlights the performance gap of LLMs in non-English languages and introduces novel recipe for synthetic instruction tuning dataset. They used 1.8 million instruction and response pairs in 51 languages bys selectively translating Orac instruction fine-tuning dataset and Tamil was one of those 51 languages. This experiment showed both Mistral 7B and Phi-3-Small performed better with synthetic instruction set. However, the experiment for done on 7B parameter models and might be expensive to fine-tune a large model and evaluation is done mainly on reasoning tasks and no evaluation is done on generation tasks like summarisation.

[Egalitarian] Identifies that the standard Byte Pair Encoding is not ideal for languages like Tamil as they divide the Tamil tokens into smaller tokens unnecessarily leading to large context window and increased computational cost and introduces a new tokenizer called Grapheme Pair Encoding (GPE). This is something worth exploring as a next step for this work.

[Chitranuvad] is multi-lingual and multimodal LLMs that is designed for translation. The model was able to achieve state of the art performance for Hindi and Malayalam language translations by employing instruction tuning and task specific fine-tuning. Although the evaluation is done only on three languages (Hindi, Malayalam and Bengali), it gives enough evidence that task specific fine-tuning works for translations.

[MOVE THIS TO LIMITATION]

[0.0.1GB] explores the use of cross lingual vocabulary adaption for non-English languages by expanding vocabulary in low resource settings. English centric models LLaMA 2 (7B), LLaMA 3 (8B) and Gemma 2 (9B) are fine-tuned then evaluated for translation and summarisation with only 30k sentences

[Fine tuning LLMs for Low Resource Lang] presents language specific modelling technique for low resource languages, more importantly it highlights the lack of evaluation and benchmark metrics for low resource languages, but no solutions are presented.

[LLMs are few shot] explores the effectiveness of LLMs as few-shot learners for low resource languages and highlights shortcomings of the in-context label alignment of low resource language and proposes query alignment as alternative

# **DESIGN AND METHODOLOGY**

This study is made up of several important steps as outlined in the Fig 1. The two models Tamil LLaMA 7B and Tamil Mistral 7B forms the foundation for this work. The datasets are created by taking explanations of these poems by Tamil scholars from the internet. To explore the performances of models with trained with more data, additional explanations were generated for the study using GPT-4o, which understands Tamil Sangam literature very well. The additional explanations were generated using few-shot learning technique. Both Tamil LLaMA and Tamil Mistral were fine-tuned separately on both manually curated dataset and the dataset that includes manually curated explanations and generated explanations. Performances of the baseline models and the 2 sets of fine-tuned models are evaluated. These steps form the framework for this study.

[FIG]

## **3.1 DATASETS**

Tamil Sangam literature corpus has 18 major texts and Thirukural. Although Thirukural is not one of these 18 texts, it is still considered as Sangam literature. Only Thirukural, Agananuru and Purananuru are selected for this study to keep the scope clear and manageable.

Thirukural contains 1330 couplets (9310 words) divided in 3 broad categories and 133 chapters, each chapters have 10 couplets. Agananuru contains 400 poems, and Purananuru contains 398 poems (2 are missing) of varying sizes and around 45000 words in total.

Although Tamil has vast collection of literature work in print media, its presence in digital media is very limited, this means the datasets needed for this study are not readily available.

Thirukural and its explanation texts were taken from Thirukural Karpom website. Thirukural has explanation written by 3 most famous Tamil scholars and all three explanations for each of the poems are collected in the dataset. Thirukural data was then cleanup by removing non-Tamil and HTML markup tokens. For Agananuru and Purananuru poems, their explanation, history and contextual data are taken from Tamil Drops website. The data was not in a structured format, and it took several iterations to extract the poems and their explanations and contextual information.

The data collected from Thirukural Karpom and Tamil Drops were manually verified to ensure correctness and quality. The manual verification was done by 2 native speakers who have at least 5 years of formal education with medium of instruction as Tamil.

Both Tamil LLaMA and Tamil Mistral are using same instruction template which is of the following format:

<s>[INST]

{poem}

[/INST]

{poem meaning and contextual information} </s>

The collected datasets are converted to this instruction format for fine-tuning. The datasets also include contextual information like who author, subject, timeline and other information like the poem number, special words with meaning – all of these are included in the poem’s meaning sections of the instruction template to provide better context for the training.

GPT-4o was used to generate additional datasets both to increase the number of records and to provide alternative explanation text for each of the poems. The reason for doing this is to increase the possibility of generalisation when fine-tuning. The generated data was manually verified by following same process followed for curated dataset to ensure both quality and correctness. The generated dataset was then concatenated to curated dataset resulted in two set of datasets – one with just the curated data and another with both curated and generated data.

We then kept 10% of the data as holdout dataset for testing and used the remaining 90% training.

## **3.2 MODELS**

Open source LLMs make research and product development in any field/area that you can imagine possible without incurring huge cost and time, usually involved in pre-training. However, the availability of models of various size and capability makes choosing a pre-trained model, for the problem at hand, a significant task. Choosing a right model could result in significant increase in performance and quality while reducing the time and cost involved in prompt engineering and fine-tuning.

To evaluate the feasibility of this task (explaining Tamil Sangam Literature in modern Tamil), we need a model that is ready to be experimented with. Because of this and the requirement of a model to work with Tamil tokens, the number of LLMs available to select for this task have been reduced significantly.

Tamil LLaMA Instruct 7B (Balachandran, 2023) and Tamil Mistral Instruct 7B (Hemanth Kumar, 2025) were chosen as models for this study because they satisfy both language and ready-to-use requirements. We also use state of the art GPT-4o, which has very good understanding of Tamil Sangam literature, for generating augmented datasets.

**Tamil LLaMA.** The Large Language Model Meta AI (LLaMA) 2 is one of the models trained and open sourced by Meta AI. LLaMA was trained using 2 trillion tokens created from publicly available information (excluding personal data) but predominantly English. LLaMA models, like GPT, are autoregressive decoder-only transformers. The chat model was fine-tuned with nearly 28,000 prompt-response pairs. Reinforcement learning with human feedback (RLHF) was used for alignment. The Tamil LLaMA which is built by enhancing the tokens of LLaMA 2 with 16,000 Tamil tokens then by fine-tuning using translated Alpaca dataset and OpenOrca data is one of the models used for this study. Tamil LLaMA is one of the good open-source Tamil LLMs, which performs reasonably well in HellaSwag and Winogrande Benchmark scores.

**Tamil Mistral.** Mistral 7B is a 7.3B parameter LLM developed by Mistral AI using transformers architecture. Mistral 7B uses a variant of the standard attention mechanism called grouped-query attention (GQA). The model’s performance is optimised by calculating attention with specific groups of hidden states rather than across all the hidden states, this also improve scalability and efficiency. Mistral 7B does not have any moderation mechanisms. Tamil Mistral which is one of the models used for this study is developed by adding 50,000 new Tamil tokens and pre-trained with 25GB of Tamil text then fine-tuned using 400,000 Tamil instruction sets.

**GPT-4o.** GPT-4o is a multilingual and multimodal GPT model developed by OpenAI. The API of GPT-4o is used in this study for augmented text generation. This model is known for its ability for summarisation, translation and answering questions. GTP-4o has a deep understanding of Tamil language and it generates very clear and coherent Tamil text, and it is also very good at taking instruction in Tamil. The model also works great when generating explanations for Tamil Sangam literature. We employed both just-ask and few-shot learning mechanisms to generate augmented explanation texts for poems used for training. [USED CHAT COMPLETION API – ADD INFO ON THIS]

## **3.3 FINE-TUNING**

Fine-tuning was done using one of the PEFT methods - QLoRA (Dettmers et al., 2024) method, which quantizes pretrained model to 4-bit and uses Low Rank Adapters (LoRA). This enables finetuning of large models in a significantly smaller GPU. Although the models are reduced to 4-bit, it still preserves the 16-bit finetuning performance. The 4-bit NormalFloat (NF4) used as quantisation type and float16 is used as compute type.

We have instruction fine-tuned both Tamil LLaMA 7B and Tamil Mistral 7B models with two sets of datasets – curated and augmented datasets, resulting 4 different models to compare.

The state-of the art NVIDIA A100, 40GB GPU in Google Colab was used for fine-tuning, each model took around 3 hours for fine-tuning.

LLaMA 2 finetuning pipeline for fine-tuning was developed loosely based on great work from Labonne, 2023.

## **3.4 HYPERPARAMETERS**

**LoRA.** The parameters below were set up for the LoRA during fine-tuning

* Rank – This parameter controls the intrinsic dimensionality of the low-rank decomposition matrix used. Rank was set to 64 as this value provides moderate complexity which enable capturing diverse pattern without overfitting, this is important as we use small datasets.
* Alpha – This parameter defines the scaling factor for learned weights. Alpha was set to 16 as it is a common choice for when rank 64 is used.
* Dropout – This is a regularisation parameter to prevent overfitting by randomly dropping units during training. This was set to 0.1 (10%), as this is a recommended rate for preventing overfitting in a model with moderate number of parameters.

**GPT-4o.** The parameters below were used for augmented text generation

* Max Completion Tokens – This parameter defines the upper bound for number of tokens that can be generated for a completion. This value was not set to allow GPT to create a very detailed description without any limitation on the size.
* Temperature – This parameter defines how creative or random the model is allowed to be and the range for this value is between 0 and 2 (inclusive). We set this value to 1.0 to have balanced focus and creativity when generating augmented text.

**Tamil LLaMA and Tamil Mistral Fine-tuning.** The parameters below were used for fine-tuning Tamil LLaMA 7B and Tamil Mistral models on Tamil Sangam literature dataset

* Number of training epochs – This parameter defines number of training epochs to perform. We set this to 10 as we found during early experiments that loss was still reducing after 5 epochs, so it is expected that setting epochs to10 would allow reducing the loss without overfitting.
* Gradient accumulation steps – This parameter defines the number of update steps to accumulate the gradients for, before performing backward/update pass. This value was set to 1, essentially disabling accumulation to allow faster weights updates and to reduce the GPU memory consumption required to store intermediate gradients.
* Optimiser – PagedAdamW (32 bit) optimiser was used as it is efficient in handling memory constraints by offloading optimiser states to CPU RAM when not in active use thereby reducing GPU memory pressure.
* Learning rate – The initial learning rate is set to 2e-4 as it balances stability and the speed of convergence for AdamW optimiser in transformer-based models.

* Weight decay – This is set to 0.001 to provide a very small regularisation which prevents overfitting but allows model to learn effectively.
* BF16 – BF16 floating-point format was used over FP16 as it is best for large-scale-models like 7B and provides wide range although compromises on precision.

* Max Gradient Normalisation – This parameter is used for gradient clipping, which limits the size of gradients during backpropagation. This is set to 0.3 to prevent large gradient updates.
* Warmup ratio – This parameter allows gradually increasing the learning rate during some initial training steps. This is set to 0.03 to reduce the risk of large initial gradient updates leading the unstable model early on.
* Group by length – This parameter indicates whether to group the sequences into batches with same length. This was enabled to save memory and speedup training.
* LR Scheduler Type – This parameter defines how learning rate is adjusted. The cosine annealing scheduler was used to as it gradually decreases learning rate following cosine curve and helps escape the local minima.
* Max Sequence Length – This is set to None to allow sequences of variable length without truncating or padding them. This combined with Group by length parameter allows optimising the memory without truncation.

**Tamil LLaMA and Tamil Mistral Inference.** The parameters below were used at inference time for Tamil LLaMA and Tamil Mistral fine-tuned models

* Max Length – This parameter defines the generated token size. We experimented with 512 and 1024 and settled on 512 as it balances the accuracy and inference speed.
* Max Return Sequence – This parameter defines number response to return. This was set to 1.
* No Repeat NGram Size – This parameter defines how to handle repeated phrases. This is set to 2 to avoid any repetitive tokens.

## **3.6 PERFORMANCE METRICS**

[SHOULD INCLUDE ROUGE OR NOT]

BLEU (Bilingual Evaluation Understudy) is a method for evaluating similarity between texts. BLEU is quick, inexpensive and language independent. It compares consecutive phrases of generated text with the consecutive phrases of reference text and counts number of matches in a weighted fashion.

Where:

Brevity Penalty

METEOR (Metric for Evaluation of Translation with Explicit Ordering) measures the quality of generated text and reference text by considering both accuracy of individual words (precision and recall) and their order within the sentence, the higher the score the better the similarity. METEOR achieves this by breaking down the text into chunks, then matching words by exact matching, stemming and synonyms, then applying penalty for out of order chunks.

Where:

chrF Score (Character-level F-Score) evaluates the similarity of reference text and generated text by comparing the overlap of character sequence (n-grams). It essentially measures how well translated text captures the character-level structure. This is suitable for this study because of the complex morphology of Tamil language. F-score is calculated using the precision and recall scores calculated from character level n-gram matches.

Where:

P – Precision

R – Recall

– Weight factor

BERTScore is used to evaluate the similarity between two pieces of text by leveraging the power of the Bidirectional Encode Representation from Transformer (BERT) model. BERT assess the semantic similarity which means the comparison is more nuanced than traditional metrics like BLEU or ROUGE which provides similarity score based on word matching. By using BERTScore, we are measuring how similar the generated text to the reference text by considering contextual meaning of the words within a sentence rather than just their presence. For reference and generated the precision, recall and F1 scores are

Where:

C = Set of token embeddings in the generated sentence

R = Set of token embeddings in the reference sentence

= Token embeddings from BERT

= Cosine similarity between token embeddings

[SHOULD I ADD FORMULA FOR ACCURACY, PRECISION AND RECALL AS WELL?]

Human Evaluation is employed to measure quality, coherence and simplicity. Reference text and generated text are shared with native speaker who had at least 5 years of formal education where medium of instruction is Tamil. As part of this study, we asked the individual to rate the generated text with scores 0 to 9 on the aspects of how similar the generate text and reference text are, does the text look coherent without any grammatical errors and whether the text is simple to follow and understand. The score 0 indicates that the generated text is nothing like reference text and it also suffer from grammatical errors, not simple to follow. The score 9 indicates that the generated text is very similar to reference text semantically and text is also coherent and simple to understand.

## **3.7 PUBLISHED MODELS AND DATASETS**

Fine-tuned models can be found at

1. <https://huggingface.co/kodebot/tamil-llama-7b-instruct-tamil-poem-fine-tune-web-only-data-attempt-1-v0.1>
2. <https://huggingface.co/kodebot/tamil-llama-7b-instruct-tamil-poem-fine-tune-web-aug-attempt-1-v0.1>
3. <https://huggingface.co/kodebot/tamil-mistral-7b-instruct-tamil-poem-fine-tune-web-only-data-attempt-1-v0.1>
4. <https://huggingface.co/kodebot/tamil-mistral-7b-instruct-tamil-poem-fine-tune-aug-data-attempt-1-v0.1>

Dataset prepared for this work can be found at

1. <https://huggingface.co/datasets/kodebot/Thirukural_tamil_with_meaning>
2. <https://huggingface.co/datasets/kodebot/Thirukural_tamil_with_meaning_aug>
3. <https://huggingface.co/datasets/kodebot/Agananuru_Tamil_with_meaning>
4. <https://huggingface.co/datasets/kodebot/Agananuru_tamil_with_meaning_aug>
5. <https://huggingface.co/datasets/kodebot/Purananuru_Tamil_with_meaning>
6. <https://huggingface.co/datasets/kodebot/Purananuru_tamil_with_meaning_aug>

# **EXPERIMENTAL RESULTS AND DISCUSSIONS**

[WHAT HARDWARE USED FOR DATA GENRATION]

[WHAT HARDWARE USED FOR FINE-TUNING AND INFERENCE]

[TODO: explain how augmented data created using GPT – just ask and few shot prompting]

We have run the inference pipeline for the validation dataset and the inference time was very good when using A100 GPUs (under 3 seconds per instruction). Optimising inference time is one of the tasks That need exploring further.

## **4.1 AUGMENTED DATASET USING GPT-4o**

Leveraged state of the art GPT-4o for generating a new dataset with explanation for Thirukural, Agananuru and Purananuru. First experimented with few-shot learning approach whereby provided 3 examples poem-explanation text pairs for each of the literature and asked GPT-4o to generate explanation text for other poems. The generated was reasonable but not satisfactory, also experimented with different temperature before settling on temperature 1 to balance both creativity and focus.

Then experimented just-ask approach by simply asking GPT-4o to explain the poem by providing the name of the literature and the poem number. GPT-4o was able to generate correct explanation text in some cases but it was completely wrong other times, may be this is because GPT-4o was not able to correctly identity the poem from the prompt provided or it doesn’t know how to identity the poem from literature name and the poem number.

Then we expanded the just-ask approach by providing the complete poem in the prompt and asking it to generate the explanation text. GPT-4o seem to have very good understanding of these poems and initial results were promising, so we used this approach to generate explanation text for all the poems in Thirukural, Agananuru and Purananuru.

To ensure the generated text is of good quality, we first cleaned up any English and non-Tamil tokens then reviewed all the explanation manually to make sure they are relevant and grammatically correct. We generated explanation text again for some of the poems (42) with different temperature (1.2) as the initial ones were not satisfactory.

The explanation texts were then formatted and converted into instruction sets that uses LLaMA 2 prompt template. These instruction sets were then merged into curated dataset to create a new dataset. This resulted in two set of datasets – curated dataset and another one that includes instructions from curated dataset and generated data.

## **4.2 FINE TUNING OF TAMIL LLAMA 2 7B**

## **4.3 FINE TUNING OF TAMIL MISTRAL 7B**

## **4.3 PERFORMANCE EVALUATION OF FINE-TUNED MODELS**

The quality of inference text was evaluated using ROUGE (Lin, 2004) score. The ROUGE evaluation score from test dataset was not great, we got 0 for all variations (rouge1, rouge2, rougeL and rougeLsum) of the score.

[TODO: bertscore]

[TODO: chrf]

[TODO: meteor]

[TODO: sacrebleu]

[TODO: manual verification]

|  |  |
| --- | --- |
| A graph of different colored bars  AI-generated content may be incorrect. | A graph of different colored bars  AI-generated content may be incorrect. |
| A graph of different colored bars  AI-generated content may be incorrect. | A graph of different colored bars  AI-generated content may be incorrect. |

However, when verified manually, the Tamil text completion (generated text) from instruct model is somewhat related to the input text, although they are not coherent, grammatically correct, contains some unexpected non-Tamil tokens and many repeated tokens or phrases, it is the evidence that this approach is in the right direction.

Although this is not a great result, but based on the understanding of how LLMs work, this is something that we can improve by obtaining larger corpus of data for fine-tuning. The bigger the corpus, the better the result would be.

# **CONCLUSIONS, FUTURE WORK AND ETHICAL ISSUES**

[TODO: Review]

[line by line meaning dataset]

[DIFFERENT TOKENISERS NOT EXPOLORED]

[DATASET IS STILL LIMITED]

[FACTUAL INFORMATION LIKE POEM NUMBERS CANNOT BE GENERALISED]

[ETHICAL ISSUE]

We did not experiment with different QLoRA parameter options due to time and compute constraints. Training was done for 10 epochs but we noticed that the training loss was not going down after 5 epochs in both Tamil LLaMA 7B and Tamil Mistral models, this is probably due to small training data size.

Also, no changes were made to tokeniser as part of this study due to time constraints but this needs to be explored further.

This study was also very limited in terms of the choice of model’s architecture. LLaMA uses decoder only architecture (Touvron et al., 2023) also known as Autoregressive models. Autoregressive models are good at text generation and a good choice of summarisation (Touvron et al., 2023), however, if we consider explainer of ancient Tamil poems in modern Tamil as a kind of translation, then decoder only models may not be a great choice and we could do better with the use of encoder. So, for translation tasks, full transformer models like BART (Lewis et al., 2019) or T5 (Raffel et al., 2020) that implements ground-breaking innovation presented in “Attention is all you need” paper (Vaswani et al., 2017) could be worth exploring.

Again, due to time constraints, in-context learning and other prompt engineering techniques were not explored for this pilot study but this is something that is worth exploring in the implementation stage.

# **REFERENCES**

Askell, A., Bai, Y., Chen, A., Drain, D., Ganguli, D., Henighan, T., Jones, A., Joseph, N., Mann, B. and DasSarma, N. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Balachandran, A. 2023. Tamil-Llama: A New Tamil Language Model Based on Llama 2. *arXiv preprint arXiv:2311.05845*.

Cui, Y., Yang, Z. and Yao, X. 2023. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Dettmers, T., Pagnoni, A., Holtzman, A. and Zettlemoyer, L. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*. **36**.

Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L. and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Kudo, T. and Richardson, J. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V. and Zettlemoyer, L. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Lian, W., Goodson, B., Pentland, E., Cook, A., Vong, C. and Teknium 2023. OpenOrca: An Open Dataset of GPT Augmented FLAN Reasoning Traces. *HuggingFace*. [Online]. [Accessed 23 February 2024]. Available from: https://huggingface.co/datasets/Open-Orca/OpenOrca.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries *In*: *Text summarization branches out*., pp.74–81.

Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation *In*: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*., pp.311–318.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*. **21**(1), pp.5485–5551.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P. and Hashimoto, T.B. 2023. Stanford alpaca: an instruction-following llama model (2023). *URL https://github. com/tatsu-lab/stanford\_alpaca*.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P. and Bhosale, S. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*. **30**.