Tiny Models to tell tiny stories in resource constrained languages

Stanford CS224N Custom Project

Niranjan Thanikachalam
Department of Computer Science
Stanford University
niranjnt@stanford.edu

Abstract

Telling convincing stories is an integral part of who we are as human beings. Stories surround us everywhere, from scientists setting the direction of academic discourse to public leaders guiding public discourse, convincing stories lie at the heart of our collective progress. In this study, we are interested in Tamil language models that can tell stories with the same complexity as told to toddlers and young children. To build such a model we created a machine translated version of the TinyStories dataset with 1M stories in the train split. We then explore GPTNeo and Llama models of differing sizes, all less than 150M parameters to learn story telling. We take a three stage approach, where the model is first pretrained on internet quality Tamil data. Next the machine translated dataset is used for continual training. Followed by this we run a final fine tuning run with a very small expert curated dataset of 2000 stories in the train split. We also attempt LoRA fine tuning of an English language GPTNeo model. We see that while the models are able to tell stories, they are not of high quality, yet we believe we have a first step in building such tiny models for low resource languages.

1 Key Information to include

- TA Mentor : Mingjian Jiang
- External collaborators : No
- External mentor : No
- Sharing project : No

2 Introduction

Story telling is a distinctly human trait. Our ability to tell convincing stories is now seen as one of the reasons Homo Sapiens Sapiens prevailed over other hominids. From prehistoric cave paintings to today's social media, story telling has played a compelling role in our shared identity. It has the power to bring us together over a shared joy as well as the power to divide us over a perceived ill. It frequently drives our actions, influences us on what we think is important, what traditions to uphold, who, what, when and how to empathize with and how not to. Critical moments in our shared history can be attributed to stories we told ourselves based on events that transpired at these moments. In a lot of ways, our stories define us. They can be anything from why one human perceives mathematics to be the ultimate truth, to why another human thinks war is the only way out of a crisis, or even why this human (I) perceived that "story-telling for an NLP project is interesting"!

Toddlers are avid story-tellers too. They make up stories about simple events around them. Articulate the stories with all the seriousness that an adult could muster and believe in the story they say to be true, as a way of justifying the immutable as well as predictable ways of the complex world around them. These stories are typically not very long. The train of thought and plot arc is usually concise and simple. Yet, being able to tell such stories still requires one to develop a consistent understanding of the world, in addition to simply understanding grammar and gaining vocabulary. While large language models are convincingly good at story telling, we are particularly interested in very small models with 150M or fewer parameters, since such models can be good candidates for inference on low compute resources. It has recently been shown that coherent, convincing story telling in English is possible in very small language models with just 80M parameters or less and that the architecture is less important than the dataset itself, in the TinyStories[1] paper. In this project, I am interested in developing a similar tiny model in a resource constrained language - Tamil, that can tell coherent, convincing stories. The goal is to develop a tiny causal language model for generating stories in Tamil when given a short prompt. An example of this is illustrated below.

Input :ஒரு நாள், லீலான்னு ஒரு சின்ன பொண்ணு அவ ரூம்ல ஒரு ஊசி ...

Output : ஒரு நாள், லீலான்னு ஒரு சின்ன பொண்ணு அவ ரூம்ல ஒரு ஊசி பாத்தா. அது கூர்மையா இருக்கும்னு அவளுக்குத் தெரியும் அதனால விளையாடக் கூடாதுன்னு தெரிஞ்சுக்குச்சு. லீலா அந்த ஊசிய அவ அம்மாகிட்ட குடுக்கனும்னு நெனச்சா, ஏன்னா அவ அம்மா அவ சட்டையில ஒரு பொத்தான் தைக்கனும். லீலா அவ அம்மா கிட்ட போயி, "அம்மா, எனக்கு ஒரு ஊசி கெடச்சுது. நீங்க இத வச்சு என் சட்டைல பொத்தான் தைக்கலாமா?"ன்னு கேட்டா. அம்மாவும் சிரிச்சுக்கிட்டே, "ஆமா லீலா, நம்ம ரெண்டு பேரும் சேர்ந்து தைக்கலாம்"ன்னு சொன்னாங்க. ரெண்டு பேரும் சேர்ந்து அந்த ஊசிய வச்சு லீலா சட்டைல பொத்தான் தைச்சாங்க. அவங்க ரெண்டு பேரும் சேர்ந்து செஞ்சதால ஈஸியா முடிஞ்சது. பொத்தான் தச்சி முடிச்சதும், லீலா அவ அம்மாவுக்கு நன்றி சொன்னா.

In this project we consider models from the GPTNeo family as was proposed in the TinyStories paper. Additionally, we also consider a tiny Llama model of similar size, inspired from SmolLM2[2]. We will use a pretrained sentencepiece tokenizer for Tamil [3]. For language pretraining, we use the Tamil split of CulturaX[4] dataset, consisting of internet quality data. We then continually train our models on a machine-translated version of the TinyStories dataset. Followed by this, we fine tune our models using a very small expert curated Tamil stories dataset. We also consider a LoRA adaptation from a pretrained 33M English TinyStories model directly. To evaluate, we report the perplexity of our models, as well as GPTEval results on the following axes: Consistency, Coherence, Grammar, Plot. We observe that while the larger models are able to generate decent quality stories, the tiny models struggle. We present evidence for dataset idiosyncracies that possibly hamper our model performance and propose better strategies. We also point to inconsistencies in GPTEval when using GPT3.5 vs GPT4o, arguing perhaps baseline values presented in the original paper where too optimistic.

3 Related Work

Small Language Models (SLMs) with <150M parameters have historically struggled to produce coherent, grammatically correct long form text with variability in content and so receive less academic focus compared to large models. Models like GPT2 with 117M parameters had difficulty generating coherent text. But recent studies show that when high quality, domain specific dataset is present, well structured learning strategies can help SLMs to achieve strong Causal language modelling capability.

In TinyStories[1] a GPT generated dataset of around 2M stories understandable by 3-4 year olds is presented. They show that with this dataset, models with as low as 33M parameters can generate coherent stories rich in plot variability. The largest model that they train is 80M parameters large and achieves very high scores on coherence, plot, consistency, grammar and instruction following axes while lagging in creativity behind GPT-4, suggesting creativity continues to improve with model size and dataset compared to the other aspects. Furthermore, in "Textbooks Are All You Need"[5] text book style web data and GPT generated data are used to train a 1.3B parameter model, resulting in significant improvement in reasoning, generalization, further confirming that dataset is as crucial as size. Kaddour

et al[6] extend this idea by showing that using a large teacher model to generate synthetic training data for smaller student models can greatly increase performance of SLMs. A similar development is SmolLM2[2], which is a 1.7B Llama type model that was trained on a synthetically generated 11 trillion token curriculum that was dynamically adjusted to address model shortcomings. SmolLM2 achieves state of the art results among similar sized SLMs, illustrating that extended training and curated data can substantially improve SLM performance. This data-centric approach taken by recent successful SLMs aligns with the idea of knowledge distillation and compression techniques as seen in DistilBERT[7], MobileBERT[8] etc.

Joshi et al[9] show that small multilingual and low resource language models have also benefited from synthetic augmentation. They report that continual pretraining of a bilingual model using machine translated text significantly improved its low resource language capabilities. Similarly, recent work on Qwen-1.5B, Phi-1.5, and Llama 3.2-1B models shows that even small-sized transformers can match or outperform much larger models if trained on well structured, high quality data.

These developments support the idea that data quality, extended training, and strategic curriculum design can enable smaller transformer based LMs to achieve strong generative performance.

4 Approach

Our primary approach is a conventional multi-stage training pipeline applied to GPT Neo and Llama models of varying sizes. In order to then test the hypothesis that in the presence of reasonable quantities of meaningful data, tiny models can still produce coherent stories, we will conduct ablation studies by removing the number of hidden layers and reducing the size of the hidden layer. We also consider two other variations by directly continually train a pretrained English Language Model.

4.1 Architecture

Tamil language has its own abugida script. For subword tokenization with Tamil inputs, we use the pretrained Tamil-LLaMA tokenizer[3], which is a sentencepiece based tokenizer derived from the LLaMA 2 English tokenizer. It was created by adding 16k tokens to the pre-existing 32k tokens, resulting in a total of 48k tokens. For Causal Language modelling, we consider GPT Neo models in 3 sizes - 152M model with 8 hidden layers and a hidden size of 1024, 68M model with 4 hidden layers and a hidden size of 786 and finally a 8M model with 8 hidden layers and a hidden size of 128. These 3 models have the same architecture as the 80M, 33M and 3M models in the original TinyStories[1] paper. However the massive increase in the tokenizer vocabulary results in an increase in the token embedding matrix size, consequently we end up with models that are roughly twice as big. We also consider a LLaMA model with a hidden size of 512, with 4 hidden layers and 8 attention heads, resulting in a 62M model, that roughly matches the mid-sized GPT Neo model. The LLaMA model was chosen to have this configuration so it can be compared with the mid-sized GPTNeo model sinse they have a similar total parameter count and similar number of hidden layers.

4.2 Multi-stage Training Pipeline

Our default method is a three stage training pipeline. First we perform pretraining on a very large internet quality Tamil language split of the CulturaX dataset [4]. Following this we train with a machine translated version of the TinyStories dataset in the second stage. In the final stage, we fine tune the model with a very small expert curated Tamil stories dataset.

4.3 Continual Training of English TinyStories Model

We also consider two variations of direct training of the English language 33M GPTNeo model using Tamil translated Tinystories dataset. To do this, we first expand the token

embedding layer (word token embeddings/wte in GPTNeo) to match the vocabulary size of the Tamil-LLaMA tokenizer and initialize it with default Xavier initialization. In the first variation, we simply continue training from this point forward, resulting in a 68M GPTNeo model. In the second variation, we freeze the rest of the model and use Low Rank Adaptation (r=8) to update the query and value projection layers in the transformer with a 10% dropout. We train this LoRA model with the Tamil Tinystories dataset.

5 Experiments

We now describe our experiments.

5.1 Data

We used the following datasets.

- CulturaX is a large internet scale multilingual dataset with a large Tamil corpus of about 3.85B tokens and will be used for pretraining. [4]
- The original TinyStories English dataset has 2.12M rows in the train split and 22k rows in the validation split. For the purpose of this project, we created a Tamil version of TinyStories by machine translation using the Google Translate API. Due to rate limits and the short window for the project, we translated the entire validation split, but only the first 1M rows in the train split. Thus the Tamil tiny stories dataset is roughly half the size as the original. We hoped that this dataset will impart world knowledge, coherence and consistency to the model.
- Tamil Stories[10] with 1.2k stories in Tamil, that was generated from children's weekly magazines sponsored by Cohere AI. We hoped that though this is an extremely small dataset, it will help impart local knowledge and cultural aspects unique to Tamil to the model.

Dataset	# Entries	Avg tokens per Entry	Purpose
CulturaX - Tamil	4.73M	816	Pretraining
TinyStories - Machine Translated	1M	234.51	Fine Tuning (coherence)
Tamil Stories	1202	891.85	Fine Tuning (local context)

Table 1: Summary of the datasets used for multistage training pipeline

5.2 Evaluation method

To measure the progress of the pretraining stage, we simply track the perplexity of the validation splits in both the Tamil TinyStories dataset and the high quality, expert curated Tamil Stories dataset. During finetuning, in addition to these two perplexity values, we also report GPT Eval values as defined in the original TinyStoies paper[1]. This works by asking a large GPT model to evaluate the generated story based on the following four criterion: Creativity, Consistency, Grammar and Plot on a scale of 0-10. We use two types of prompts for the tiny story generation:

1 **Type A prompts** The first 50 characters of a fixed, but randomly selected set of 50 stories from the Tamil Tinystories dataset. These prompts are very similar to the kind of prompts the model saw during training.

2 **Type B prompts** I personally wrote 67 single sentence prompts in Tamil, closely resembling how Tamil stories naturally begin, using Tamil names and places, covering a range of topics like pets, nature, family, entertainment, mythical beings and feelings.

These prompts can be found in the code submissions in "evaluation/validation.json" and "evaluation/test.json" respectively. Furthermore we evaluate these against two different GPTs: GPT 3.5 Turbo and GPT4o. The exact system prompt for GPTEval is shown in appendix

5.3 Experimental details

We used the Huggingface Trainer for training all the models, except the base model (pretraining) of the GPTNeo 152M. This model alone was trained using Pytorch Lightning. We used AdamW with default betas $(\beta_1, \beta_2) = (0.9, 0.999)$ for pretraining and $(0.9, 0.95)$ for fine tuning. Weight decay was set to 0.01 during pretraining and 0.1 during final fine tuning. We used gradient accumulation of 160 batches and a batch size of 8 for the 152M model. We used gradient accumulation of 80 batches and a batch size of 16 for all the other models. Gradient norm clipping was set to the default value of 1.0. Linear LR scheduler with warmup was used in all cases, with the LR set to 5e-4 during pretraining, 1e-4 during first stage of fine tuning and 1e-5 during the final stage. Pretraining ran for 2 epochs and lasted 4 days for the 152M model and 3.5 days for the 68M model on a RTX4090. First stage finetuning was run for 10 epochs and generally lasted for 40-50 hours depending on the model and final stage finetuning takes about 10 minutes. All models are available for use from huggingface hub and a list will be made available in the appendix.

5.4 Results

Model	Perplexity TinyStories	Perplexity Tamil Stories
GPT Neo 152M Base	16.16	89.92
GPT Neo 68M Base	19.14	37.97

Table 2: Perplexity values for the TinyStories and the high quality expert curated TamilStories datasets obtained during pretraining of the GPTNeo 152M and GPTNeo 68M models.

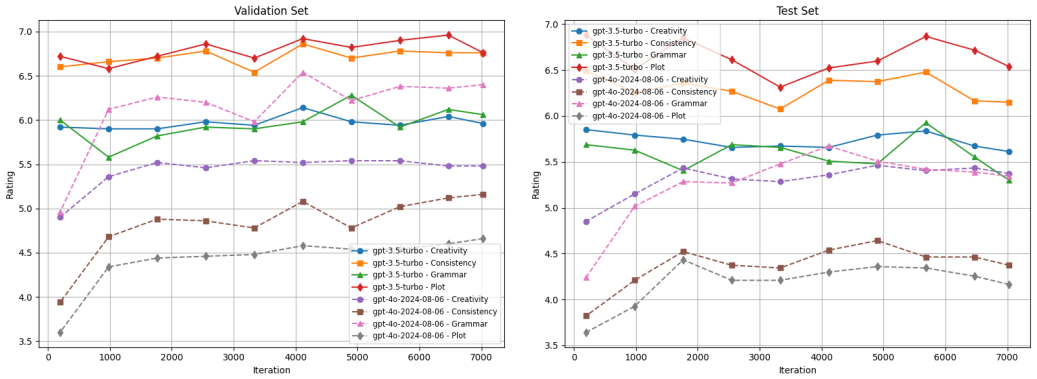
Model	Perplexity TinyStories	Perplexity Tamil Stories	Creativity	Consistency	Grammar	Plot
GPT Neo 152M FineTuned	4.07	2111.97	5.94 5.93	6.74 6.69	5.84 5.81	6.78 6.69
GPT Neo 68M FineTuned	4.22	1197.95	6.02 5.88	6.70 6.22	5.84 5.43	6.62 6.85
Llama 62M FineTuned	4.89	2234.9	5.94 5.60	6.54 5.94	5.96 4.82	6.88 6.93
GPT Neo 8M FineTuned	7.32	3199.8	5.62 5.46	5.96 5.78	5.26 4.79	5.96 6.30
GPT Neo 152M FineTuned HQ	N/A	74.07	5.70 5.10	6.14 5.34	4.94 4.61	6.70 6.22
GPT Neo 68M Direct Trained	4.27	3349.11	5.94 5.82	6.70 6.58	5.88 6.07	6.78 6.68
GPT Neo 68M LoRA	109.72	12629.53	N/A	N/A	N/A	N/A

Table 3: This table lists perplexity as well as GPT Eval values (scale of 10) for the 4 criterea : Creativity, Consistency, Grammar and Plot. Evaluation is performed using GPT3.5 Turbo. For each model, the top row of GPTEval values correspond to Type A prompts (generated) while the second row corresponds to Type B prompts (user written).

Table 2. presents the perplexity values of the two GPTNeo models 152M,68M for both the finetuning datasets. As expected, both the perplexity values are low at the end of training and the perplexity of the high quality data is higher than that of the synthetic data. Tables 3 and 4 present the GPT Eval ratings on a scale of 10 for the 4 criterea : Creativity, Consistency, Grammar and Plot for evaluations performed using GPT 3.5 turbo

Model	Creativity	Consistency	Grammar	Plot
GPT Neo 152M FineTuned	5.40 5.39	4.92 4.57	6.26 5.46	4.48 4.36
GPT Neo 68M FineTuned	5.50 5.27	4.90 4.30	6.12 4.82	4.52 4.18
Llama 62M FineTuned	5.32 5.21	4.34 4.27	5.46 4.64	4.34 4.27
GPT Neo 8M FineTuned	4.96 4.7	3.92 3.64	3.78 3.49	3.86 3.37
GPT Neo 152M FineTuned HQ	4.67 4.9	3.55 3.82	3.25 3.58	3.38 3.6
GPT Neo 68M Direct Trained	5.24 5.24	4.52 4.24	5.7 4.92	4.28 4.04

Table 4: This table lists GPT Eval values (scale of 10) for the 4 criterea : Creativity, Consistency, Grammar and Plot. Evaluation is performed using GPT4o. For each model, the top row of GPTEval values correspond to Type A prompts (generated) while the second row corresponds to Type B prompts (user written).



(a) Evolution of GPTEval values during training for the GPTNeo 152M model, evaluated using both the GPT models for prompts of Type A

(b) Evolution of GPTEval values during training for the GPTNeo 152M model, evaluated using both the GPT models for prompts of Type B

Figure 1: Evolution of GPTEval values during finetuning stage 1 for the GPTNeo 152M model. It can be seen that the ratings are consistently higher with GPT3.5 Turbo than with GPT4o. Also, for the same evaluator, the Type A prompts score slightly better than Type B prompts

and GPT4o respectively. For each model, the top row of GPTEval values correspond to Type A prompts (generated) while the second row corresponds to Type B prompts (user written). From the tables it is clear that while we dont perform very well, we are not too bad either. It is also seen that the GPT eval values are higher in TypeA prompts than Type B prompts, indicating that training data alignment with the evaluation prompts could lead to higher evaluation scores. The evolution of these values for the GPTNEo 152M model is shown in figure 1. More figures are present in the appendix. It can also be seen that there is not much difference in performance between the 152M and 68M GPTNeo models as well as the LLaMA 62M models. Only the 8M GPTNeo model performs significantly worse.

Due to resource constraints, we were only able to fine tune the 152M model on the expert curated data, but we notice that this model performs poorly in GPTEval compared to the 152M model at the end of the first stage of fine tuning. We believe this is because the high quality TamilStories dataset is not necessarily aimed at toddlers, and therefore has more complex sentences. This perhaps confounds the model during training. The fact that first stage of finetuning increases the perplexity on the TamilStories dataset for both the 152M and 68M model as compared with the pretraining perplexity, also points to the idea that the model moves closer to "baby language" during fine tuning with TinyStories and away from complex language.

Due to resource constraints, we had to end the LoRA finetuning since convergence was slower than full finetuning. Furthermore, as shown in table 5, the average evaluation scores of 50 samples from the validation set reveal that both GPTs rate the stories poorly. Moreover, the ratings obtained by our models are competetive with these values in Table 5.

Aspect	**80M English**	**GPT3.5 Tamil Type A full**	**GPT-4 Tamil Type A full**
Creativity	7.05	6.16	5.72
Grammar	8.43	6.8	7.12
Consistency	8.98	6.1	8.18
Instruct	8.18	N/A	N/A
Plot	7.29	7.1	5.72

Table 5: English Tiny Stories GPTEval scores for a model with 1024 hidden size and 12 layers (132M equivalent). We also present the GPTEval scores of the full translations of the Type A prompts as an indication of the data quality itself.

6 Analysis

Here is a story generated by the 152M model, translated to English using Google translate for the sake of easy interpretability, along with the prompt in italics.

In a town, there was a little boy named Bob who lived with his mother. He used to play`with his toys, draw with his crayons, and wait for his snack. But his mother was busy today. She was working too hard to wait for her. She looked at her watch and said, "Sorry, Bob. You have to go to work with me. Let's do your chores." Bob was sad and bored. He didn't want to do his chores no matter what. He decided to wait, but he didn't want to go for a walk. He said, "No, Mom. I don't want to go. It's boring. I don't like my mom's chores. It's too hard." He ran to his mother's room and locked the door. His mother was sad and angry. She told Bob to stay inside and put on his shoes and coat. She gave him a backpack and a toy rack, thinking that it would make him a big boy and make him very strong. Maybe he would know when it was time to work. She said, "Boy, you come with me`

Below is the same prompt with the 68M GPTNeo model.

There was a little boy named Bob who lived with his mother in a small town. He `liked to help her with her cooking once a day. He used the pots and pans and the spoons and the plates. He also took a big jug of soup and poured it into`

```
the pot. He started stirring, stirring, stirring. It was funny to see the
water and food flying everywhere. But then he heard a knock on the door. He
opened it, and he caught sight of a tall woman with a big smile and a big
backpack. He said she had a cake. Her name was Lily. "Hello, Bob. Can I
see you Bob?" the tall woman asked. Bob was surprised, but Lily took his
hand. She saw that she had a dirty refrigerator. He opened it, and Lily saw
a cake. She wanted to try some. She tried to see what was inside, but Bob
refused. "No, no. You can't have my cake. It's mine, not yours. Go away!"
Bob said. Lily felt sad. She went back to her house.
```

While the stories dont make perfect sense, they are not complete gibberish either, indicating the evaluation by GPTEval is probably correct. As indicated by Table 5, the quality of machine translation seems to be very very poor. This perhaps has been passed on to the models trained by us. We know that state-of-the-art machine translators for Tamil[11] exists. We tried using this, but given the GPU requirement and the short time frame for this project this could not be used.

Also to consider is that our tokenizer is not fully utilized since it is bilingual working for English and Tamil. The increased model size is actually not justified and could also contribute to poor performance.

7 Conclusion

We were able to train multiple models for telling stories in Tamil. Resource and time constraints meant the data used wasn't of a high quality. This breaks the primary hypothesis that a large good quality data is more important than model size. Nonetheless, we do observe that four models in the 60M-152M range have similar performance, while the small 8M model has worse off performance. Both of these observations obey the main hypothesis. Finally, instead of pretraining with noisy internet data, perhaps it would make more sense to distill a state of the art model like RomanSetu[12] that work well for Tamil to a small model and get started from there. A second approach is to train English and Tamil together, which has become the standard way of training low-resource languages. Finally, we donot have the RLHF component now common in training CausalLMs, which also possibly contributed to the lower performance.

References

[1] Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english?, 2023.

[2] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. Smollm2: When smol goes big – data-centric training of a small language model, 2025.

[3] Abhinand Balachandran. Tamil-llama: A new tamil language model based on llama 2, 2023.

[4] Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. CulturaX: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4226–4237, Torino, Italia, May 2024. ELRA and ICCL.

[5] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck,

Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023.

[6] Jean Kaddour and Qi Liu. Synthetic data generation in low-resource settings via fine-tuning of large language models, 2024.

[7] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

[8] Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. Mobilebert: a compact task-agnostic bert for resource-limited devices, 2020.

[9] Raviraj Joshi, Kanishk Singla, Anusha Kamath, Raunak Kalani, Rakesh Paul, Utkarsh Vaidya, Sanjay Singh Chauhan, Niranjan Wartikar, and Eileen Long. Adapting multilingual llms to low-resource languages using continued pre-training and synthetic corpus, 2024.

[10] AI Tamil Nadu. Tamil stories, https://huggingface.co/datasets/aitamilnadu/tamil_stories. Accessed: 2025-02-12.

[11] Jay Gala, Pranjal A Chitale, A K Raghavan, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar M, Janki Atul Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M Khapra, Raj Dabre, and Anoop Kunchukuttan. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *Transactions on Machine Learning Research*, 2023.

[12] Jaavid Aktar Husain, Raj Dabre, Aswanth Kumar, Jay Gala, Thanmay Jayakumar, Ratish Puduppully, and Anoop Kunchukuttan. Romansetu: Efficiently unlocking multilingual capabilities of large language models via romanization, 2024.

A Appendix

A.1 Prompt for GPTEval

The following prompt was used for GPTEval.

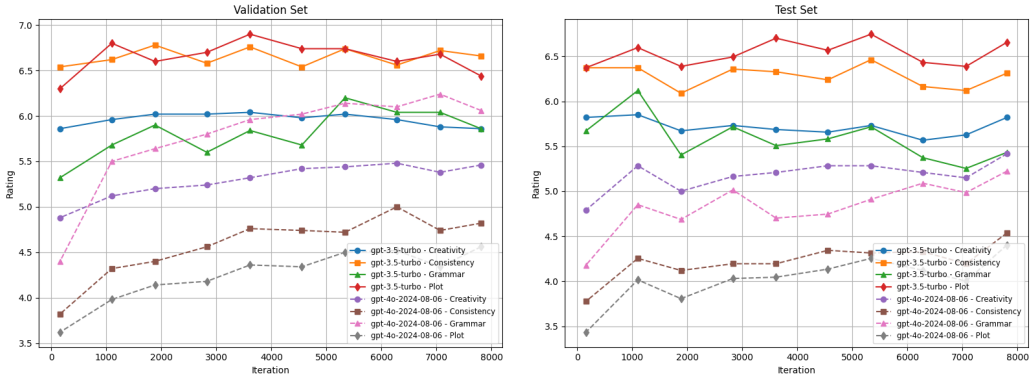
```
system_message = """You are a strict story evaluator.
    Always return exactly four numbers for the following story in this order: Creativity,
    Consistency,Grammar,Plot.  Format example: 7,8,9,6
    Do not add any extra text or explanations."""

    user_prompt = f"""
    Rate the {language} story below on a scale of 0-10 for Creativity, Consistency, Grammar,
    and Plot, keeping in mind it is to be understood by 4 year olds.

    {story}

    Respond with one line per story in the format: Creativity,Consistency,Grammar,Plot.
    """
```

A.2 GPTEval evolution for GPTNeo 68M

A.3 GPTEval evolution for Llama 62M

A.4 More story generations:

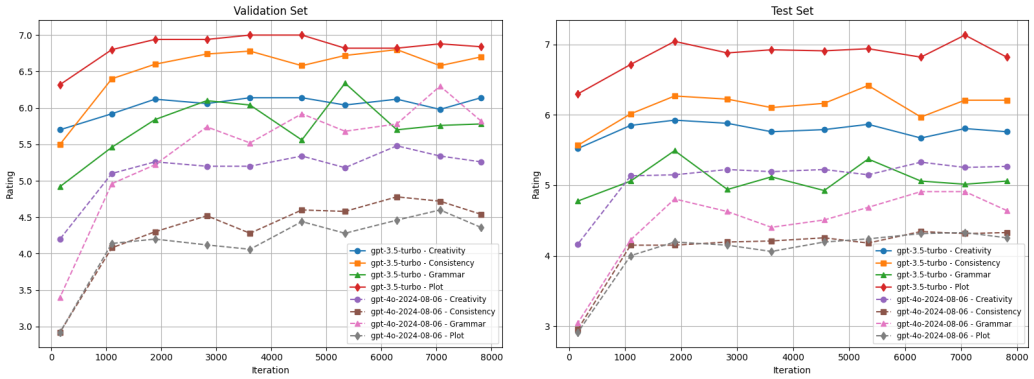
GPTNeo 152M *There lived a hare and a lion in a forest. They both wanted* to get a new animal one day. "This is a deer. It is big and brown and has a long nose. It is very fast and strong. We can take it apart and see it in a special way. Maybe it will be our friend." But before they could trust the other animals, there was a big fire in the forest. It had sharp teeth



(a) Evolution of GPTEval values during training for the GPTNeo 68M model, evaluated using both the GPT models for prompts of Type A

(b) Evolution of GPTEval values during training for the GPTNeo 68M model, evaluated using both the GPT models for prompts of Type B

Figure 2: Evolution of GPTEval values during finetuning stage 1 for the GPTNeo 68M model. It can be seen that the ratings are consistently higher with GPT3.5 Turbo than with GPT4o. Also, for the same evaluator, the Type A prompts score slightly better than Type B prompts



(a) Evolution of GPTEval values during training for the Llama 62M model, evaluated using both the GPT models for prompts of Type A

(b) Evolution of GPTEval values during training for the LLaMA 62M model, evaluated using both the GPT models for prompts of Type B

Figure 3: Evolution of GPTEval values during finetuning stage 1 for the LLaMA 62M model. It can be seen that the ratings are consistently higher with GPT3.5 Turbo than with GPT4o. Also, for the same evaluator, the Type A prompts score slightly better than Type B prompts

and yellow eyes, and it would make noise. "Wow, this is a deer. It is very beautiful and cute. But you should leave it alone. It might be afraid of us or hurt us." The hare and the lion ran deeper into the forest. They watched the deer for a while, and then decided to leave it alone. They looked for something else to do. They saw a big hole in a tree. It looked like a castle. They decided to explore it. They climbed up and looked into the hole. They saw a cozy cave. It was dark and mysterious. They saw many beds and a big bed. They looked around and heard a noise. "We are a deer! This is our friend

GPTNeo 68M *A hare and a lion lived in a forest. One day they saw a* squirrel on a branch. They were very hungry, and the squirrel wanted some for their food. They asked the lion if he could give the rabbit some nuts. "No, you can't take my nuts. They are nuts for my food. They are very nice, if you have good nuts," said the squirrel. "Then you should act like a hare and

be my friend. These nuts are good for you," said the squirrel. The hare felt stupid and ashamed when he saw the squirrel. He didn't listen to the squirrel, he wished they could play together. He wanted to make another nut and a friend. But the hare didn't think the squirrel would be such a stupid idea. The squirrel ate the nut, and the squirrel shook its claws, squealed and made a noise. "Look, I'm a hare! I can move my paws as I want, and I love the forest," said the squirrel. But the hare didn't want to play. He looked at the squirrel and said, "No, you can't take my nut, it's not yours. It's mine."

A.5 Model Registry

- GPTNeo 152M : tniranjan/finetuned_gptneo-base-tinystories-ta_v3
- GPTNeo 68M : tniranjan/finetuned_tinystories_33M_pretrained_tinystories_ta
- GPTNeo 68M Direct: tniranjan/finetuned_tinystories_33M_tinystories_ta
- LLaMA 62M: tniranjan/finetuned_Llama_tinystories_tinystories_ta
- GPTNeo 8M: tniranjan/finetuned_tinystories_3M_tinystories_ta