NepaliGPT 2.0: Nepali Text Understanding and Generation

Prince Kumar Singh
princesingh@princelab.org

Rajeeb Thapa Chhetri
rajeevthapa@princelab.org

Abstract

Large Language Models (LLMs), such as ChatGPT and GPT-4o, have revolutionized the landscape of natural language processing research, hinting at the tantalizing prospect of Artificial General Intelligence (AGI). However, the formidable costs associated with training and deploying these models pose significant barriers to transparent and accessible academic research. While community-driven open-source models like LLaMA have emerged, their primary focus remains English corpora, leaving other languages underserved. In this paper, we propose an ingenious augmentation for LLaMA, empowering it to understand and generate Nepali text. Experimental results underscore the prowess of our novel model in understanding and generating Nepali content.

1 Introduction

The emergence of Large Language Models (LLMs) has significantly impacted natural language processing (NLP). These models, characterized by their vast size and extensive training data, excel in both understanding and generating human-like text. Unlike pre-trained language models (PLMs) such as BERT, which focus on text understanding, the GPT series (including ChatGPT and GPT-4) emphasize creativity and text production. Researchers continually explore techniques to reduce training costs while maintaining performance, and deployment considerations include model compression, parallel computation, memory scheduling, and structural optimization. Looking ahead, LLMs will continue to evolve, finding applications in various domains, shaping the future of NLP.

Building on this success, GPT-4 (OpenAI, 2023), GPT-4o (OpenAI, 2024) emerged, showcasing even greater potential in natural language understanding, generation, and multi-modal reasoning. These models have sparked interest in Artificial General Intelligence (AGI) and inspired research across various NLP tasks. Their impressive performance, adaptability, and few-shot learning capabilities continue to drive advancements in sentiment analysis, machine translation, and question-answering systems.

However, as impactful as LLMs have been, their implementation comes with inherent limitations that hamper transparent and open research. A major concern is their proprietary nature, which restricts access to the models, thus inhibiting the broader research community's ability to build upon their successes. Furthermore, the vast computational resources necessary for training and deploying these models present a challenge for researchers with limited resources, further compounding the accessibility problem.

To address these challenges, the NLP research community has embraced open-source alternatives, emphasizing transparency and collaboration. LLama-3.1, Qwen-2 and Mistral stand out as notable examples of such initiatives. These open-source large language models (LLMs) aim to facilitate academic research, accelerate progress in NLP, and create robust, versatile LLMs suitable for diverse applications.

In spite of the significant progress achieved by LLama and Qwen in the field of NLP, their native support for Nepali language tasks remains limited. Their vocabularies encompass only a small number of Nepali tokens, which significantly hampers their ability to process and generate Nepali text. To facilitate efficient training and deployment, we adopt the Low-Rank Adaptation (LoRA) approach , allowing us to fine-tune the models without excessive computational overhead. Our preliminary study aims to bolster LLama understanding and generation capabilities, serving as a stepping stone for researchers adapting these models to other languages, especially those that are commonly spoken in Nepal and South Asia. An overview of the proposed model is given in figure 1.

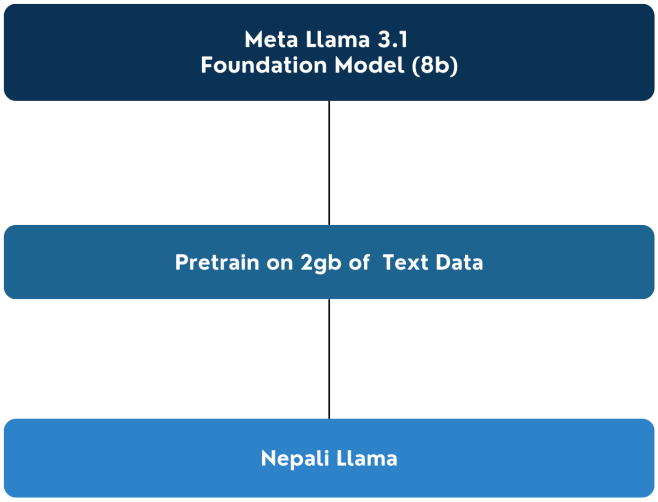

Figure 1: Overview of proposed Nepali llama

2 Nepali Llama

2.1 Background

LLaMA 3.1 is a foundational, decoder-only large language model built upon the transformer architecture. Similar to the GPT series and other transformer-based LLMs, LLaMA consists of an embedding layer, multiple transformer blocks, and a language model head. LLaMAv 3.1 is available in four different model sizes: 8B, 70B, 405B.

LLaMA has been pre-trained with a standard language modeling task using a mix of publicly available sources, such as crawled web pages, books, Wikipedia, and preprint papers. Experimental findings reveal that LLaMA delivers competitive performance compared to other LLMs like GPT-4o, albeit at a smaller model size. This compactness and effectiveness have garnered considerable attention from researchers, leading to the widespread use of LLaMA-based models.

2.3 PARAMETER EFFICIENT FINE-TUNING WITH LORA

The conventional training paradigm that updates the full parameters of LLMs is prohibitively expensive and is not time- or cost-feasible to most labs or companies. Low-Rank Adaptation (LoRA) is a parameter-efficient training method that maintains the pre-trained model weights while introducing trainable rank decomposition matrices. LoRA freezes the pre-trained model weights and injects trainable low-rank matrices into each layer. This approach significantly

reduces total trainable parameters, making it feasible to train LLMs with much less computational resources.

To be specific, for a linear layer with weight matrix $W_0 \in \mathbb{R}^{d \times k}$, where k is the input dimension, and d is the output dimension, LoRA adds two low-rank decomposed trainable matrices $B \in R^{d \times r}$ and $A \in R^{r \times k}$, where r is the predetermined rank. The forward pass with input x is given by the following equation,

$$h = W_0 x + \Delta W x = W_0 x + BAx, B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times d}$$

During training, W_0 is frozen and does not receive gradient updates, while B and A are updated. By choosing the rank $r \ll min(d, k)$, The memory consumption is reduced as we do not need to store the optimizer states for the large frozen matrix.To achieve parameter-efficient training while adhering to a tight budget, we apply LoRA training to Nepali Llama, including both the pre-training.

2.4 PRE-TRAINING OBJECTIVE

We pre-train the Nepali LLaMA model with the standard Causal Language Modeling (CLM) task. Given an input token sequence $x = (x_0, x_1, x_2, \ldots)$, the model is trained to predict the next token xi in an autoregressive manner. Mathematically, the objective is to minimize the following negative log-likelihood:

$$L_{CLM}(\Theta) = \mathbb{E}_{x \sim \mathcal{DPT}}\left[\sum_i logp(x_i | x_0, x_1, ..., x_i - 1; \Theta)\right]$$

where, Θ represents the model parameters, DPT is the pre-training dataset, xi is the token to be predicted, and $x_0, x_1, \ldots, x_{i-1}$ constitute the context.

3 Experimental Setups

3.1 Experimental Setups for Pre-training

We initialize the Nepal LLaMA model with the original LLaMA weights and conduct pre-training using fp16 on the 7B model. We directly apply LoRA to attentions and MLPs.

The models are trained on A100 GPUs (40GB VRAM) for one epoch. The parameter-efficient training with LoRA is performed with PEFT library . We employ the AdamW optimizer with a

peak learning rate of 5e-5 for linear layers and 1e-5 for token and embed layers and 5% warm-up linear scheduler for 1 epoch.

Detailed hyperparameters for each Nepali LLaMA model are listed in Table 1.

Table 1. Pre-training hyperparameters for Nepali LLaMA

Settings	Llama 3.1 8b
Training data	2gb
Batch size	64
Peak learning rate	5e-5(for linear layers)
Max sequence length	2048
LoRA rank	8
LoRA alpha	32
LoRA weights	QKVO, MLP
Trainable parameters (%)	11.9767

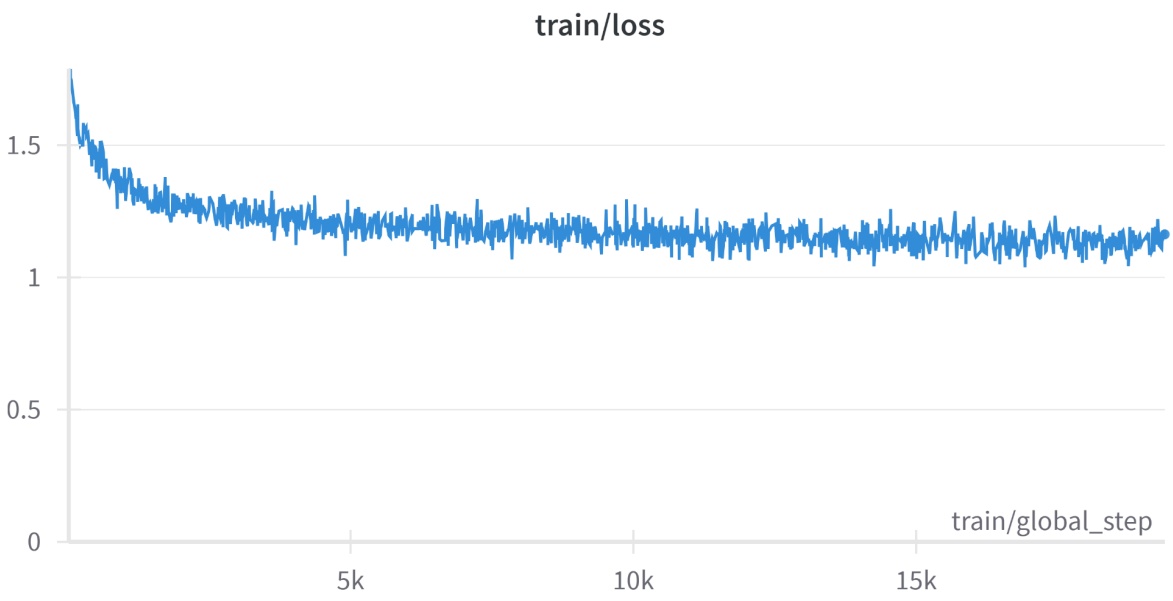

Figure 2: Training loss for pretraining

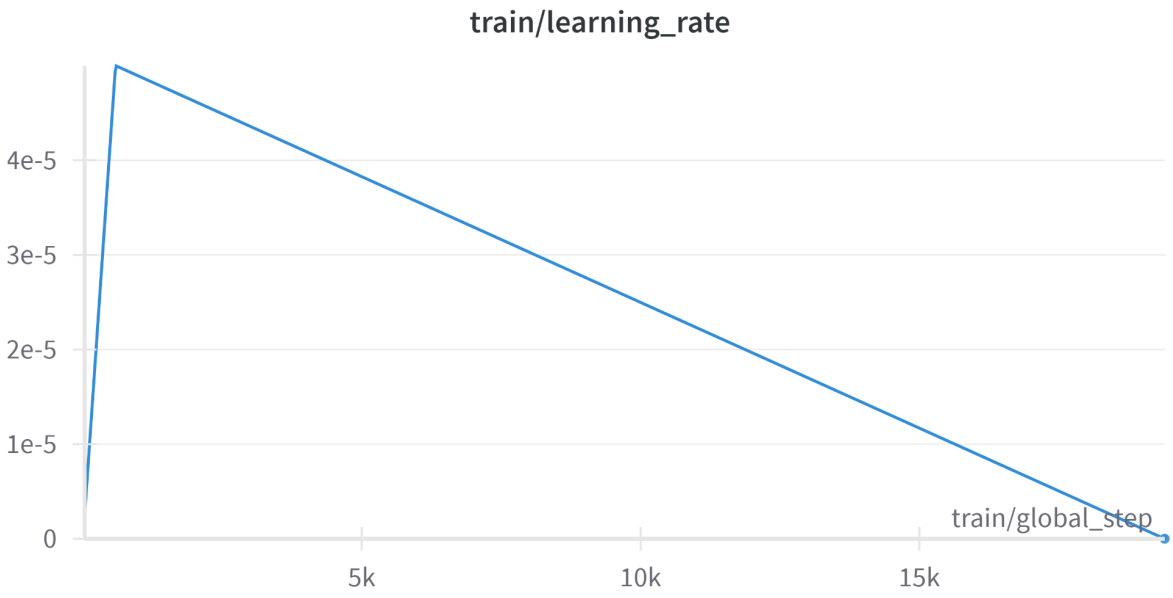

Figure 3: Learning rate

4 Conclusion

In this technical report, we have presented an approach to enhance the Nepali understanding and generation capabilities of the LLaMA model. Acknowledging the limitations of the original LLaMA's model, we pretrained it with 2gb of Nepali data.

In our future work, we will explore Reinforcement Learning from Human Feedback (RLHF) or Reinforcement Learning from AI Instructed Feedback (RLAIF) to better align model outputs with human preferences. Additionally, we plan to adopt advanced quantization techniques, including GPTQ and AWQ, and investigate alternative methods to LoRA for more efficient pre-training and fine-tuning of large language models. These efforts aim to enhance model performance and applicability across diverse tasks in the Nepali NLP community.

Limitations

1. **Harmful and Unpredictable Content**:
    - Despite our model's ability to reject unethical queries, it may still generate harmful or misaligned content due to biases in training data or limitations in discerning appropriate outputs.
2. **Insufficient Training**:
    - Due to constraints in computing power and data availability may result in suboptimal performance for the models' Nepali understanding capabilities.
3. **Lack of Robustness**:
    - The models can exhibit brittleness, producing inconsistent or nonsensical outputs when faced with adversarial inputs or rare language phenomena.
4. **Comprehensive Evaluation**:
    - Evaluating large language models remains crucial. Existing benchmarks should be thoroughly studied and adapted for LLMs to shape future research effectively.
5. **Scalability and Efficiency**:
    - While we've made the model more accessible through techniques like LoRA and quantization, combining it with the original LLaMA can still pose deployment challenges, especially for users with limited computational resources.

Future work should address these limitations to further enhance the models' capabilities, making them more robust, accessible, and effective for a broader range of applications in the Global and Nepalese NLP community.

References

1. Xiaoran Liu, Hang Yan, Shuo Zhang, Chenxin An, Xipeng Qiu, Dahua Lin.Scaling Laws of RoPE-based Extrapolation. [2310.05209] Scaling Laws of RoPE-based Extrapolation (arxiv.org)
2. Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In International Conference on Learning Representations, 2019. URL https://openreview.net/forum?id= Bkg6RiCqY7.
3. OpenAI. GPT-4 Technical Report. arXiv e-prints, art. arXiv:2303.08774, March 2023. doi: 10. 48550/arXiv.2303.08774
4. Unsloth. blog.Unsloth - 4x longer context windows & 1.7x larger batch sizes, 2024
5. Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen.LoRA:Low Rank Adaptation for Large Language model. [2106.09685] LoRA: Low-Rank Adaptation of Large Language Models (arxiv.org).
6. Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. arXiv preprint arXiv:2305.14314, 2023.
7. Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman ´ Castagne, Alexandra Sasha Luccioni, Franc¸ois Yvon, Matthias Gall ´ e, et al. Bloom: A 176b- ´ parameter open-access multilingual language model. arXiv preprint arXiv:2211.05100, 2022
8. Llama3.1. blog: Introducing Llama 3.1: Our most capable models to date (meta.com)