Llama-3-Nanda-10B-Chat:
An Open Generative Large Language Model for Hindi

Anonymous ACL submission

Abstract

Developing high-quality large language models (LLMs) for moderately resourced languages presents unique challenges in data availability, model adaptation, and evaluation. We introduce *Llama-3-Nanda-10B-Chat*, or *Nanda* for short, a state-of-the-art Hindi-centric instruction-tuned generative LLM, designed to push the boundaries of open-source Hindi language models. Built upon Llama-3-8B, *Nanda* incorporates continual pretraining with expanded transformer blocks, leveraging the Llama Pro methodology. A key challenge was the limited availability of high-quality Hindi text data; we addressed this through rigorous data curation, augmentation, and strategic bilingual training, balancing Hindi and English corpora to optimize cross-linguistic knowledge transfer. With 10 billion parameters, *Nanda* stands among the top-performing open-source Hindi and multilingual models of similar scale, demonstrating significant advantages over many existing models. We provide an in-depth discussion of training strategies, fine-tuning techniques, safety alignment, and evaluation metrics, demonstrating how these approaches enabled *Nanda* to achieve state-of-the-art results. By open-sourcing *Nanda*, we aim to advance research in Hindi LLMs and support a wide range of real-world applications across academia, industry, and public services.

1 Introduction

Recent advances in large language models (LLMs) have significantly transformed natural language processing (NLP), enabling impressive reasoning and instruction-following capabilities. However, most development has remained English-centric. While multilingual LLMs such as Falcon (Almazrouei et al., 2023), PaLM (Chowdhery et al., 2022), Bloom (Scao et al., 2022), Aya (Ustun et al., 2024), and Llama-3.1 (Dubey et al., 2024) attempt to broaden linguistic coverage, their pretraining continues to rely heavily on English-dominated corpora, ultimately limiting their performance in underrepresented languages.

In particular, although Hindi is the fourth most-spoken language globally[1,2], it is still underrepresented in existing LLMs, which demonstrate a considerable drop in performance as compared to that in English (Jin et al., 2024; Hasan et al., 2024). To address this, unlike massively multilingual models such as Bloom (Scao et al., 2022) and Aya (Ustun et al., 2024), we argue for the development of bilingual LLMs that excel on an under-represented language–which is Hindi in our case–but maintain high-performance on English. In this paper, we introduce *Llama-3-Nanda-10B-Chat* (*Nanda*), a 10B-parameter decoder-only bilingual LLM tailored for Hindi. This setup allows *Nanda* to develop natural language capabilities from the large-scale English data, and also extend these capabilities into Hindi through cross-lingual transfer.

Building high-quality Hindi LLMs presents challenges due to limited data availability (Joshi et al., 2020). In contrast to English, which benefits from corpora of up to 15 trillion tokens (Tang et al., 2024), Hindi resources are scarce. To mitigate this, we curated a 65B token Hindi corpus for continual pretraining and developed a data processing pipeline to ensure high-quality data, which includes code-mixed (with English) and romanized Hindi examples. We also prepare a set of ∼81K instructions across both Hindi and English, spanning over several diverse set of NLP tasks. We use an equal ratio of Hindi-English tokens for pretraining and apply oversampling during instruction-tuning to balance the 64.5M English and 43.5M Hindi tokens in the instruction-tuning dataset.

Nanda builds on Llama-3 (Dubey et al., 2024), incorporating recent breakthroughs such as

[1] `https://en.wikipedia.org/wiki/Hindi`

[2] `https://www.worlddata.info/languages/hindi.php`

RoPE (Su et al., 2021) and grouped-query attention (Ainslie et al., 2023), along with a custom-built tokenizer for bilingual optimization. We evaluate the model across Hindi and English benchmarks in reasoning, factuality, safety, bias and generation. Results show that *Nanda* is one of the best-performing Hindi-English bilingual language model, achieving competitive results in reasoning and factuality tasks while outperforming similarly sized models in text generation. These results mark a promising step toward robust, high-quality LLMs for Indic languages.

2 Pretraining Data Preparation

Nanda is pre-trained on billions of words to build a strong foundation in Hindi, with a knowledge base tailored to language's cultural nuances. We curated a large pre-training dataset by incorporating diverse Hindi-language sources, including websites, news articles, books, and , Wikipedia. This dataset integrates resources such as Hindi-specific datasets from HuggingFace, IIT-Bombay English-Hindi Parallel Corpus (Kunchukuttan et al., 2018) and High Performance Language Technologies' Multilingual Datasets (Burchell et al., 2025) (see Appendix A for details). The pretraining data is described in appendix In total, our pre-processed dataset comprises of **65 billion tokens of Hindi data**.

2.1 Preprocessing Pipeline

We perform a comprehensive pre-processing step on our pre-training data to ensure that *Nanda* learns from diverse, high-quality data. Here, we provide a brief outline of the workflow of our pre-processing pipeline, which is also illustrated in Figure 1.

Detokenisation A large portion of the raw data in our pre-training corpus comes from publicly available datasets, some of which are already pre-processed or tokenised. To ensure consistency, we *detokenise* the raw data, standardizing the texts across all datasets. At this stage, all documents in our corpus are non-tokenised regardless of their original source.

Filtering Following detokenisation, we filter out irrelevant and low-quality documents using several heuristics as follows: *short content removal*, where documents with less than 20 words are removed; *long word removal*, where documents containing words longer than 100 characters (URLs or gibberish strings) are removed; *Hindi sentence threshold*, where we ensure that at least 50% of the sentences in each document are in Hindi (using fastText lid.176.bin); *Hindi character threshold*, where we ensure that at least 70% of the characters in each document are in Hindi; *special symbol removal*, where documents with more than 20% of their characters as special symbols, punctuation or numerical digits are removed.

Cleaning We further refine the dataset by cleaning the filtered documents using the following techniques: *Unicode fix*, where we repair corrupted Unicode sequences; *normalization*, where we standardize Hindi punctuation and character forms across the dataset; *HTML/JS removal*, where we remove HTML or JavaScript tags and scripts from each document; *citation removal*, where we remove citations to maintain the text's coherence and logical flow; *boilerplate removal*, where we remove repetitive boilerplate text from each document to reduce redundancy; *bad word removal*, where we detect and remove inappropriate and offensive words / phrases from each document; *noisy n-gram removal*, where we detect and remove meaningless or repetitive n-gram patterns from each document. We use ftfy, pydantic, nltk and spaCy to perform cleaning.

Deduplication Finally, we leverage locality-sensitive hashing (MinHash) to perform *deduplication* on the remaining documents. Ultimately, the size of the final pre-processed dataset is reduced to 42% of the total raw text in the original data sources.

Developing the pre-processing pipeline for Hindi posed greater challenges as compared to English. While English pre-processing pipelines benefit from numerous large-scale, open-access datasets, and well-established techniques, Hindi requires a custom-built approach. Insights gained from experiments with smaller LLMs and the pre-processing pipeline for the pre-training dataset used for *Jais* (Sengupta et al., 2023) guided the selection of heuristics used in the final pipeline for *Nanda*'s pre-training dataset. However, due to the limited availability of Hindi data, we applied less aggressive filtering than most approaches, ensuring that valuable Hindi content was retained. Details of token counts after each pre-processing step are included in Appendix B.

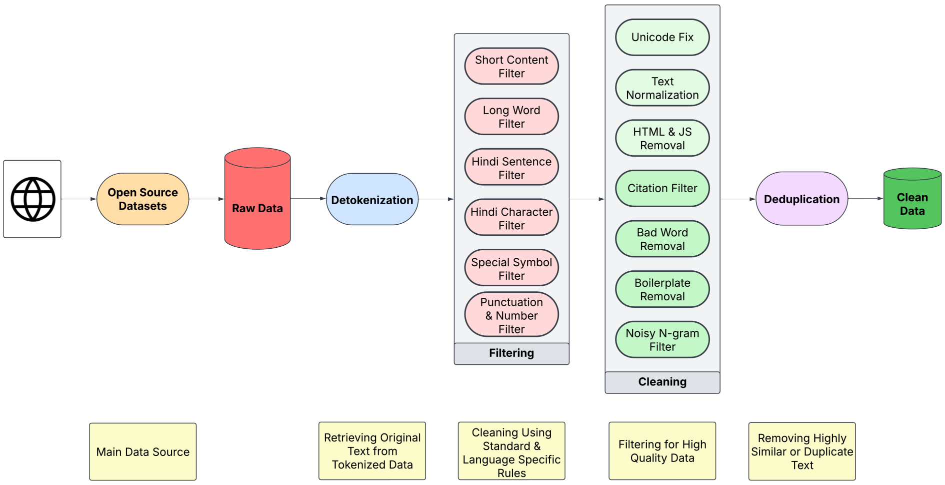


Figure 1: Our Hindi preprocessing pipeline. We pre-process the raw text through a series of steps. We perform filtering using several heuristics, clean the filtered documents using various techniques, and finally perform deduplication to get the final pretraining corpus.

2.2 Mixing Hindi and English Data

During the adaptation of the Llama-3 model, we mix Hindi and English data following the findings of Gosal et al. (2024): continual pre-training of a foundation model on new, previously unseen data can help in bilingual adaptation. However, when this new language/domain data is out-of-distribution of the original training data, it can cause forgetting of prior capabilities, which is referred to as a stability gap. To mitigate forgetting, we can incorporate a small amount of replay data, closer in distribution to the original pre-training data (Guo et al., 2024). In our work, we conduct extensive experiments to determine the minimum proportion of English replay data that should be mixed with Hindi to maintain prior capabilities, while also learning the new language.

For adapting Llama-3-8B for Hindi, our experiments revealed that a relatively high amount of replay data is necessary. Specifically, we found that a 1:1 English-to-Hindi dataset mix worked best, enabling cross-lingual capability transfer while preventing saturation of Llama-3 for domain adaptation. For replay data, we used a mix of textbook, mathematics, coding and reasoning datasets from publicly available sources.

3 Model

3.1 Tokenizer and Architecture

***Nanda* Tokenizer** The first step in adapting a foundation model for multilingual use is to construct a balanced vocabulary that includes all target languages. Recent state-of-the-art models such as Llama-3 (Dubey et al., 2024) use byte pair encoding tokenizers (Sennrich et al., 2016), primarily trained on English data.

These tokenizers often split non-English words into characters or bytes, creating a significant imbalance among languages. This imbalance introduces inefficiency in pretraining, fine-tuning and inference. A balanced multilingual tokenizer with low fertility (Rust et al., 2021) in all languages offers three main advantages: (i) lower training and inference cost; (ii) reduced latency during inference; and (iii) longer context windows (Petrov et al., 2023). Furthermore, models trained with low-fertility tokenizers tend to perform well on downstream tasks (Ahuja et al., 2023).

In Table 1, we show that the Llama-3 tokenizer requires up to 2.6× more tokens than words when applied to Hindi text, indicating significant inefficiency. To address this, we introduce ***Llama-3-ExtVocab-20***, a 20K Devanagari-enriched extension of the Llama-3 tokenizer, which reduces the fertility score to 1.19, a 54.4% decrease. This is achieved by incorporating the most frequent Hindi tokens from our pretraining corpus, while ensuring that none of the added tokens overlap with the original vocabulary.

We conduct a vocabulary extension analysis to determine the optimal number of new Hindi tokens to be added, ensuring a balanced multilingual vocabulary. The Hindi tokens are borrowed from a monolingual Hindi tokenizer trained on the Hindi corpora. We create a few candidate extended vocabularies and perform intrinsic evaluations following

Ali et al. (2024). For intrinsic evaluation, we use the fertility score to measure the efficiency of the tokenization process (Gosal et al., 2024). Fertility is defined as $f = \frac{S}{W}$, where S is the total number of tokens in the tokenized text and W is the number of words in the raw text. It is important to note that fertility is calculated on the held-out subsets of the Hindi corpora, which were not used for tokenizer training.

Table 1 shows the intrinsic evaluations of three candidate tokenizers, (i) ***Llama-3-ExtVocab10***, (ii) ***Llama-3-ExtVocab20***, and (iii) ***Llama-3-ExtVocab30***, which extend the Llama-3 vocabulary by 10%, 20%, and 30%, respectively. Based on our tokenizer fertility ablation studies, *Llama-3-ExtVocab20* reduces the fertility of Llama-3's tokenizer by 54.40% while maintaining fertility in English. It achieves a fertility score of 1.19 on Hindi, which is comparable to the base Llama-3 tokenizer's English fertility of 1.35. Extending the vocabulary to 30% shows minimal improvement in Hindi fertility, therefore, we select ***Llama-3-ExtVocab20*** as the tokenizer for *Nanda*.

***Nanda* Embeddings** Following the methods outlined for embedding initialization in Gosal et al. (2024), we use a semantic similarity search-based embedding initialization method. This method uses Wechsel multilingual initialization (Minixhofer et al., 2022) where pretrained embeddings like Fasttext or OpenAI embeddings are used. For each new Hindi token added to the Llama-3 base vocabulary, we identify top-k most similar tokens in the base vocabulary based on cosine similarity using embeddings from a pretrained embedding model. We use OpenAI's `text-embedding-3-large` embeddings (Kusupati et al., 2024) for its superior quality and multilingual capabilities. To initialize the embeddings of the new Hindi token, we take a weighted average of the top-k similar tokens' base embeddings. After experimenting with different values for the k, we achieve the best results with $k = 5$. This initialization method was used for embedding and unembedding layers of *Nanda*.

***Nanda* Architecture** Recently, decoder-only models have achieved state-of-the-art performance in generative language tasks. *Nanda* is derived from Llama-3 8B (Dubey et al., 2024) leveraging the Llama-Pro approach (Wu et al., 2024); hence, it has the standard causal decoder-only transformer architecture. Building upon the Llama-3 model, we incorporated both recent advances from the literature and insights from our own experiments. Following Wu et al. (2024), we leverage the block expansion approach, which proves to be highly effective for language adaptation, especially for low-resource languages. By adding and fine-tuning additional decoder blocks initialized to identity mappings while freezing the original Llama-3 backbone, we train only the newly added blocks.

This enables the model to integrate new domain and language-specific knowledge without forgetting previously learned information. Although the techniques described in Wu et al. (2024) focus on code and math adaptation, we successfully adapted this approach for language adaptation. We start with Llama-3-8B base model and expanded the number of decoder blocks from 32 to 40 using an interleaved approach. A new decoder block was added every 4 decoder blocks in the base Llama-3 model. In our language adaptation experiments, we found that an optimal data mix of 1 : 1 (En:Hi) yielded the best results (in downstream 0 shot tasks in both English and Hindi) compared to Hindi-only adaptation. In both experiments, we trained on a total of 55B tokens for Hindi in order to maintain the same token count for the appropriate comparison. Our results show that the block-expansion approach is a strong candidate for language adaptation with less training overhead and resources compared to training domain-specific models from scratch, especially for low-resource languages. In the future, this work could expand to other architectures (like Mixture-of-Experts) and modalities, and it would be interesting to analyse the impact on overall accuracy in downstream tasks. Following the results from Gosal et al. (2024), we find that the optimal adapter layers are 25% of the existing layers.

3.2 Pre-Training

Nanda uses a 40-layer architecture with 32 attention heads and a hidden dimensionality of 4096. For optimization, we use a peak learning rate of 1.5e-4 and a batch size of 4 million tokens. For the continual pretraining dataset, we sampled documents from the source list described in Section 2 and generated sequences with a context length of 8,192 tokens. When a document was smaller than 8,192 tokens, it was concatenated with other document (documents) and packed into one sequence. `<|endoftext|>` is used to demarcate the end of each document, giving the language model the information necessary to infer that tokens separated

	Llama-3	**ExtVocab10**	**ExtVocab20**	**ExtVocab30**
Vocab Size	128,256	141,081	153,856	166,732
Hindi Fertility	2.61	1.27 (-51.34%)	**1.19 (-54.40%)**	1.16 (-55.55%)
English Fertility	1.35	1.35	1.35	1.35

Table 1: Tokenizer intrinsic evaluation across vocab sizes. Adding Hindi vocab reduces fertility by 51.34%, 54.40%, and 55.55% in *ExtVocab10*, *ExtVocab20*, and *ExtVocab30*, respectively, compared to the base Llama-3 tokenizer.

by `<|endoftext|>` are unrelated.

We train *Nanda* using the AdamW optimizer (Loshchilov and Hutter, 2018) with $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 1e-5$, and weight decay of 0.1. We scale the gradient norms using a maximum norm clipping value of 1.0. The learning rate schedule comprises a linear warm-up to peak learning rate for 1% of the total steps, followed by a 10× cosine decay for the rest of the steps. After packing, we used a global batch size of 4M tokens. All training, hyperparameter tuning, and instruction-tuning experiments were conducted on the Condor Galaxy 2 AI supercomputer from Cerebras[3] (see Appendix E for details on the training infrastructure).

4 Instruction-Tuning

An effective LLM must accurately interpret user instructions across diverse NLP tasks and adhere to their preferences for helpfulness & safety. However, pretraining alone does not enable *Nanda* to accurately interpret and respond to user instructions. To address this, we instruction-tune (Ouyang et al., 2022) the pre-trained model using a high-quality instruction dataset, aligning the model for practical use-cases and enhancing safety in its responses.

4.1 Dataset

Nanda is developed as a bilingual model, and thus, it must be enabled to understand instructions in Hindi without compromising its performance in English. To this end, we prepare a diverse dataset containing ∼81K instructions (Hindi and English) in a prompt-response pair format over a diverse set of NLP tasks including safety-alignment.

English Instructions The English subset of our instruction-tuning dataset comprises ∼39K high-quality instructions spanning a comprehensive range of tasks. In particular, we have close to 20K instructions focused on mathematics, while the rest of the examples cover code and various types of reasoning, such as physical, logical and causal reasoning. Formatted into prompt-response pairs, this subset consists of 7.7M tokens in prompts and 9M tokens in their responses, adding up to a total of ∼17M tokens.

Hindi Instructions As a relatively low-resource language, Hindi does not have many high-quality instruction-tuning datasets. Several existing approaches have utilized machine translation on subsets of English instruction-tuning datasets to create datasets for low-resource languages. We create our Hindi instruction-tuning dataset using a similar technique; selecting a set of publicly available English instructions focusing on various forms of reasoning, and translating it into Hindi using various machine-translation models. We realize that Hindi speakers often use a more relaxed form of the language during informal interactions. We aim for our model to be adept at understanding both formal and informal writing styles. So, we translate the English instructions into two forms of written Hindi:

- **Formal Hindi –** The translated instances are written in Devanagari script with a style of writing consistent with official documents in Hindi. This was done using the Google Translate API (2024-12 snapshot).

- **Casual Hindi –** Generated translations contain Hindi (and some English) words using a mix of Devanagari and Latin scripts. This form of the language is generally used by Hindi-speaking individuals during informal conversations like texting, informal speech, interactions on social media, *etc*. This was done using GPT-4 (OpenAI, 2023a).

Subsequently, several Hindi language experts ensure the quality of translations by manually verifying a sample of instances from the generated dataset. Ultimately, the Hindi instruction-tuning subset comprises ∼22K high-quality machine-translated Hindi instructions, split into ∼13.5K in

[3] *Introducing Condor Galaxy 1: A 4 ExaFLOPS Supercomputer for Generative AI* – Cerebras

formal Hindi and the remaining in casual Hindi. In particular, this subset comprises 3.8M prompt tokens and 10M response tokens, or a total of ∼14M tokens.

Safety-Tuning Data We developed a comprehensive safety prompt collection process specifically tailored for Hindi model training, covering eight types of attacks and over 100 detailed safety categories. In the current released version, we randomly sampled 20K data for SFT (see Appendix F for more details).

4.2 Instruction-Tuning Setup

As mentioned in Section 4.1, the instances in our raw instruction-tuning data contain a system instruction and a pair of a user-prompt and an AI response. In the case of multi-turn interactions, we have a sequence of multiple prompt–response pairs. Since our model is built on top of *Llama-3-8B-Instruct*, we templatize each raw datapoint using the *Llama-3-Instruct* prompt template both for supervised fine-tuning (SFT) and for inference.[4] At this stage, we oversample the instructions in our dataset (excluding safety instruction-tuning data) to 300% of the original quantity to strengthen the model. This means we perform SFT over approximately 100M tokens consisting of 47M tokens in Hindi instructions and 53M of the same in English instructions. Moreover, similar to *Jais* (Sengupta et al., 2023), we apply padding to each templatized instance, use the same autoregressive objective as for pretraining, and mask the loss of the prompt to make sure backpropagation considers only the answer tokens during SFT.

5 Evaluation

In this section, we aim to provide a thorough assessment of the *Nanda* model across a diverse set of evaluation dimensions, covering downstream NLP tasks, safety assessments, and generation capabilities. These evaluations are designed to rigorously measure the model's performance and adaptability, particularly in supporting multilingual use cases across both Hindi and English languages.

5.1 Downstream Evaluation

Evaluation Setup We conduct a comprehensive downstream evaluation, comparing *Nanda* model to a series of baselines that support both Hindi and English languages. Our baseline models include models that are specifically optimized for the Hindi language, such as Gajendra-v0.1 (BhabhaAI, 2024), Nemotron-4-Mini-Hindi (Joshi et al., 2024), Airavata (Gala et al., 2024) and models from the AryaBhatta series (GenVRadmin, 2024a,b). We also include multilingual models such as Aya-23 (Aryabumi et al., 2024) and Mistral (Mistral AI, 2024). Additional models include popular general-purpose models like Llama-3, Llama 3.1, and the latest Llama-3.2 (Dubey et al., 2024).

We adopt the LM-Evaluation-Harness framework (Gao et al., 2021) to evaluate each model in a zero-shot setting and report the accuracy for each task. Within the framework, the context string is concatenated with each candidate output string, and the answer is determined by selecting the concatenated string with the highest normalized log-likelihood.

We perform the comparative evaluation of *Nanda* against other LLMs for both Hindi and English, building upon the evaluations conducted in prior studies (Dubey et al., 2024; Aryabumi et al., 2024; OpenAI, 2023b).

For each language, our evaluation encompasses aspects such as knowledge, reasoning, and misinformation, as outlined in Table 2 and Table 3. For Hindi, we assess performance on four translated benchmarks—MMLU-hi, HellaSwag-hi, ARC-hi, and TruthfulQA-[MC1,MC2]-hi that are fetched from Okapi[5] (Dac Lai et al., 2023). For English, following prior studies, we include MMLU (Hendrycks et al., 2020), HellaSwag (Zellers et al., 2019), ARC (Clark et al., 2018a) and TruthfulQA-[MC1,MC2] (Lin et al., 2021).

Results for Hindi Table 2 presents the zero-shot evaluation results for Hindi. *Nanda* demonstrates superior performance across many evaluation criteria, placing itself among the state-of-the-art Hindi language models. Specifically, compared to Indic models, such as Gajendra-v0.1, Airavata, AryaBhatta series models, *Nanda* achieves significant absolute improvements across knowledge retrieval, commonsense reasoning and misinformation. We can further see that among multilingual models, Llama-3.1 and Aya-23-8B are among the best-performing models, with an average accuracy of 38 and 36, respectively. However, *Nanda* outperforms both of them by 2.68 and 4.94 absolute points.

[4] https://www.llama.com/docs/model-cards-and-prompt-formats/meta-llama-3/

[5] https://huggingface.co/alexandrainst

Model	Average	MMLU-hi 0-shot	HellaSwag-hi 0-shot	ARC-hi 0-shot	TruthfulQA-MC1-hi 0-shot	TruthfulQA-MC2-hi 0-shot
Airavata-7B	0.3204	0.3044	0.3287	0.2551	0.2600	0.4540
Gajendra-v0.1-7B	0.2949	0.3028	0.3304	0.2594	0.2096	0.3723
AryaBhatta-GemmaOrca-8.5B	0.3712	0.3682	0.4191	0.3022	0.2962	0.4701
AryaBhatta-GemmaUltra-8.5B	0.3858	0.3900	0.4394	0.3168	<u>0.3027</u>	0.4801
Nemotron-4-Mini-Hindi-4B-Instruct	**0.4103**	<u>0.4294</u>	<u>0.4772</u>	**0.3579**	<u>0.3027</u>	<u>0.4841</u>
Aya-23-8B	0.3602	0.3350	0.4481	0.2971	0.2820	0.4390
Mistral-7B-Instruct-v0.3	0.3435	0.3069	0.3435	0.2637	**0.3053**	**0.4981**
Meta-Llama-3-8B	0.3752	0.4010	0.4340	0.3280	0.2630	0.4500
Meta-Llama-3-8B-Instruct	0.3804	0.3850	0.4070	0.3360	0.2930	0.4810
Llama-3.1-8B-Instruct	0.3828	0.4290	0.4500	0.3310	0.2620	0.4420
Llama-3.2-3B-Instruct	0.3518	0.3660	0.3860	0.2920	0.2690	0.4460
Llama-3-Nanda-10B-Chat	<u>0.4096</u>	**0.4299**	**0.4922**	<u>0.3476</u>	0.2975	0.4810

Table 2: Evaluation results on **Hindi** benchmarks. *Average* represents the mean score across tasks, and *0-shot* indicates zero-shot results. For all columns, higher the better. **Bold** represents the best scores in that column while <u>underlined</u> represents the second-best scores.

Model	Average	MMLU 0-shot	HellaSwag 0-shot	ARC-en 0-shot	TruthfulQA-MC1 0-shot	TruthfulQA-MC2 0-shot
Airavata-7B	0.4470	0.4044	0.6798	0.4448	0.2607	0.4070
Gajendra-v0.1-7B	0.4422	0.3955	0.7308	0.4311	0.2521	0.4062
AryaBhatta-GemmaOrca-8.5B	0.5406	0.5195	0.7370	0.4551	0.3880	0.5406
AryaBhatta-GemmaUltra-8.5B	0.5465	0.5374	0.7573	0.4893	0.3660	0.5465
Nemotron-4-Mini-Hindi-4B-Instruct	0.5359	0.5528	0.7122	0.4893	0.3513	0.5021
Aya-23-8B	0.4924	0.4474	0.7431	0.4525	0.3035	0.4924
Mistral-7B-Instruct-v0.3	**0.6167**	0.5898	**0.8318**	**0.5885**	**0.4211**	0.5966
Meta-Llama-3-8B	0.5526	0.6134	0.7942	0.5338	0.2742	0.5526
Meta-Llama-3-8B-Instruct	0.5911	0.6369	0.7598	0.5689	0.3599	0.5911
Llama-3.1-8B-Instruct	0.5988	**0.6644**	0.7939	0.5500	0.3696	0.5988
Llama-3.2-3B-Instruct	0.5338	0.5878	0.7083	0.4577	0.3244	0.4970
Llama-3-Nanda-10B-Chat	<u>0.6096</u>	<u>0.6499</u>	<u>0.8022</u>	<u>0.5776</u>	<u>0.3995</u>	**0.6190**

Table 3: Evaluation results on **English** benchmarks. *Average* represents the mean score across tasks, and *0-shot* indicates zero-shot results. For all columns, higher the better. **Bold** represents the best scores in that column while <u>underlined</u> represents the second-best scores.

Nemotron-4-Mini-Hindi is the best performing model, outperforming *Nanda* as per average accuracy on log-likelihood evaluations. However, we observe that *Nanda* outperforms Nemotron-4-Mini-Hindi on generation evaluation in Hindi and English (see Section 5.2) by a significant margin. This highlights the need for comprehensive and more holistic model evaluations to better understand its performance and capabilities.

Results for English We also conducted an evaluation for English, with the results shown in Table 3. Notably, *Nanda* achieves a slight improvement over existing English models. Additionally, we observe that, apart from the AryaBhatta series and Nemotron-4-Mini-Hindi model, other Hindi models, such as Gajendra-v0.1 and Airavata, exhibit significantly lower performance than established English models.

5.2 Generation Evaluation

In addition to downstream and safety evaluations, we also assess the models' core capability for Hindi text generation. Consistent with prior studies (Peng et al., 2023; Vicuna, 2023), we adopt an LLM-as-a-judge evaluation methodology using GPT-4o (OpenAI, 2023b). The evaluation is based on the *Vicuna-Instructions-80* (Vicuna, 2023) dataset[6], which was manually translated into Hindi by professional translators to ensure linguistic fidelity.

We generate model responses to the Hindi prompts from the *Vicuna-Instructions-80* dataset, using a temperature of 0.3 and a repetition penalty of 1.2. As baselines, we compare against open-source multilingual models such as Llama-3-8B-Instruct (Dubey et al., 2024) and Nemotron-4-Mini-Hindi-4B-Instruct (Joshi et al., 2024) (Nemotron-Hi-4B-Instruct).

GPT-4o serves as the evaluator, scoring each pair of outputs on a scale from 0 to 10 based on quality, relevance, and fluency in Hindi (see Appendix D for our evaluation prompt).

Our generative evaluation results, summarized in Figure 2, show that *Nanda* significantly outperforms all baselines in Hindi text generation. Built upon the Llama-3 (8B) architecture, *Nanda* retains

[6] https://lmsys.org/blog/2023-03-30-vicuna/

Model	English	Hindi
Airavata-7B	57.95	55.97
Gajendra-v0.1-7B	44.03	39.02
Aya-23-8B	49.48	63.79
AryaBhatta-GemmaOrca-8.5B	62.88	58.14
AryaBhatta-GemmaUltra-8.5B	61.55	50.47
Llama-3.1-8B-Instruct	**90.99**	87.01
Llama-3-Nanda-10B-Chat	85.97	**87.96**

Table 4: Evaluation results for Safety (% queries where the generated response was safe). **Bold** represents the best scores for that language

efficiency while introducing improvements that enhance its alignment with the Hindi language, as illustrated in Figure 2:b. Furthermore, *Nanda* surpasses Nemotron-Hi-4B-Instruct, demonstrating superior contextual understanding and generating more natural and fluent Hindi text in language-focused tasks.

5.3 Safety Evaluation

Following previous work (Wang et al., 2023a), we constructed a novel dataset for Hindi safety evaluation, aiming to identify biases and harmful content within the language model, specifically focused on Hindi and its cultural context. The evaluation results from over 1056 risky questions are shown in Table 4. We can see that our model achieves similar safety performance to Llama-3.1-8B-Instruct and is much safer than the other models. Please refer to Appendix F for more details.

6 Related Work

Multilingual language models have evolved from English-centric pre-training (Devlin et al., 2019a; Radford et al., 2019; Raffel et al., 2023; Biderman et al., 2023) to monolingual models in other languages (Faysse et al., 2024; Gutiérrez-Fandiño et al., 2022; Zeng et al., 2021; Sengupta et al., 2023; Phan et al., 2022; Koto et al., 2020; Ko et al., 2023) and multilingual training across a few or many languages (Nguyen et al., 2024; Mesham et al., 2021; Ogueji et al., 2021; Jude Ogundepo et al., 2022; Xue et al., 2021; Chung et al., 2023; Shliazhko et al., 2023; Scao et al., 2022; Lin et al., 2022; Conneau et al., 2020; Khanuja et al., 2021; Oladipo et al., 2023; Alabi et al., 2022; Dabre et al., 2022). Models like mT5 (Xue et al., 2021) and umT5 (Chung et al., 2023), trained on the mC4 corpus, offer broad language coverage but primarily rely on unsupervised pre-training and require downstream fine-tuning for specific tasks. Another line of work focuses on expanding language support post hoc through methods such as continued fine-tuning or vocabulary expansion (Yong et al., 2023; Luukkonen et al., 2023; Lin et al., 2024b; Imani-Googhari et al., 2023), though these approaches often struggle to scale efficiently. While models such as mBERT (Devlin et al., 2019b), XLM-R (Conneau et al., 2020), and Bloom (Scao et al., 2022) include Hindi, the underrepresentation of Hindi content limits their zero-shot performance relative to monolingual models (Li et al., 2023). In contrast to prior work that emphasizes either pre-training or task-specific fine-tuning, our work focuses on enabling instruction-following capabilities in pre-trained multilingual models, allowing them to generalize across tasks without the need for downstream tuning. Appendix I presents a more comprehensive discussion of related work.

7 Conclusion

We have introduced *Nanda*, a new state-of-the-art Hindi-English bilingual instruction-tuned large language model (LLM). It can perform a wide range of generative and downstream language tasks in both Hindi and English, ranging from common-sense reasoning to natural language understanding tasks such as sentiment analysis, irony detection, and hate speech detection. Its pre-trained and fine-tuned capabilities outperform all known open-source Hindi models of similar size and are comparable to state-of-the-art open-source English models that were trained on larger datasets. We encourage researchers, hobbyists, and enterprise developers alike to experiment with and develop on top of our model, particularly those working on multi-lingual and/or non-English applications.

Nanda represents an important evolution and expansion of the Hindi NLP and AI landscape. This Hindi model, which was born in the UAE, represents an important strategic step for government and commercial organizations towards the digital revolution. By advancing Hindi language understanding and generation, empowering local players with sovereign and private deployment options, and nurturing a vibrant ecosystem of applications and innovation, this work supports a broader strategic initiative of digital and AI transformation to usher in an open, more linguistically inclusive, and culturally-aware era.

Limitations

Nanda is trained on publicly available data, including curated Hindi data, and efforts have been made to reduce unintentional biases in the dataset. However, some biases might still be present, as with all language models. Designed as an AI assistant for Hindi and English, its purpose is to enhance human productivity. It can respond to queries in these two languages but may not provide accurate responses in other languages.

The current version of *Nanda* is not finetuned for generative tasks, such as summarization, translation, and transliteration (STT). We plan to curate a suitabale STT dataset for finetuning and extensive testing in the future.

References

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. 2023. MEGA: Multilingual evaluation of generative AI. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267, Singapore. Association for Computational Linguistics.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebr'on, and Sumit K. Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *ArXiv*, abs/2305.13245.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pre-trained language models to african languages via multilingual adaptive fine-tuning. *Preprint*, arXiv:2204.06487.

Mehdi Ali, Michael Fromm, Klaudia Thellmann, Richard Rutmann, Max Lübbering, Johannes Leveling, Katrin Klug, Jan Ebert, Niclas Doll, Jasper Buschhoff, Charvi Jain, Alexander Weber, Lena Jurkschat, Hammam Abdelwahab, Chelsea John, Pedro Ortiz Suarez, Malte Ostendorff, Samuel Weinbach, Rafet Sifa, and 2 others. 2024. Tokenizer choice for LLM training: Negligible or crucial? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3907–3924, Mexico City, Mexico. Association for Computational Linguistics.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Kelly Marchisio, Sebastian Ruder, Acyr F. Locatelli, Julia Kreutzer, Nick Frosst, Phil Blunsom, Marzieh Fadaee, A. Ustun, and Sara Hooker. 2024. Aya 23: Open weight releases to further multilingual progress. *ArXiv*, abs/2405.15032.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *ArXiv preprint*, abs/2108.07732.

BhabhaAI. 2024. Gajendra-v0.1. `https://huggingface.co/BhabhaAI/Gajendra-v0.1`. Accessed: 2024-10-29.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. *Preprint*, arXiv:2304.01373.

Laurie Burchell, Ona de Gibert, Nikolay Arefyev, Mikko Aulamo, Marta Bañón, Pinzhen Chen, Mariia Fedorova, Liane Guillou, Barry Haddow, Jan Hajič, Jindřich Helcl, Erik Henriksson, Mateusz Klimaszewski, Ville Komulainen, Andrey Kutuzov, Joona Kytöniemi, Veronika Laippala, Petter Mæhlum, Bhavitvya Malik, and 16 others. 2025. An expanded massive multilingual dataset for high-performance language technologies. *Preprint*, arXiv:2503.10267.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *ArXiv preprint*, abs/2107.03374.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, and 48 others. 2022. Palm: Scaling language modeling with pathways. *Preprint*, arXiv:2204.02311.

Hyung Won Chung, Noah Constant, Xavier Garcia, Adam Roberts, Yi Tay, Sharan Narang, and Orhan Firat. 2023. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining. *Preprint*, arXiv:2304.09151.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018a. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018b. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv preprint*, abs/1803.05457.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. 2024. Or-bench: An over-refusal benchmark for large language models. *Preprint*, arXiv:2405.20947.

Raj Dabre, Himani Shrotriya, Anoop Kunchukuttan, Ratish Puduppully, Mitesh Khapra, and Pratyush Kumar. 2022. Indicbart: A pre-trained model for indic natural language generation. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics.

Viet Dac Lai, Chien Van Nguyen, Nghia Trung Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan A Rossi, and Thien Huu Nguyen. 2023. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. *arXiv e-prints*, pages arXiv–2307.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2023. Towards leaving no indic language behind: Building monolingual corpora, benchmark and models for indic languages. *Preprint*, arXiv:2212.05409.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 510 others. 2024. The llama 3 herd of models. *ArXiv*, abs/2407.21783.

Manuel Faysse, Patrick Fernandes, Nuno M. Guerreiro, António Loison, Duarte M. Alves, Caio Corro, Nicolas Boizard, João Alves, Ricardo Rei, Pedro H. Martins, Antoni Bigata Casademunt, François Yvon, André F. T. Martins, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2024. Croissantllm: A truly bilingual french-english language model. *Preprint*, arXiv:2402.00786.

Jay Gala, Pranjal A. Chitale, Raghavan AK, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar, Janki Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M. Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *Preprint*, arXiv:2305.16307.

Jay Gala, Thanmay Jayakumar, Jaavid Aktar Husain, Aswanth Kumar M, Mohammed Safi Ur Rahman Khan, Diptesh Kanojia, Ratish Puduppully, Mitesh M. Khapra, Raj Dabre, Rudra Murthy, and Anoop Kunchukuttan. 2024. Airavata: Introducing hindi instruction-tuned llm. *arXiv preprint arXiv: 2401.15006*.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation.

GenVRadmin. 2024a. Aryabhatta-gemmaorca-merged. `https://huggingface.co/GenVRadmin/AryaBhatta-GemmaOrca-Merged`. Accessed: 2024-10-29.

GenVRadmin. 2024b. Aryabhatta-gemmaultra-merged. `https://huggingface.co/GenVRadmin/AryaBhatta-GemmaUltra-Merged`. Accessed: 2024-10-29.

Gurpreet Gosal, Yishi Xu, Gokul Ramakrishnan, Rituraj Joshi, Avraham Sheinin, Zhiming, Chen, Biswajit Mishra, Natalia Vassilieva, Joel Hestness, Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Onkar Pandit, Satheesh Katipomu, Samta Kamboj, Samujjwal Ghosh, Rahul Pal, Parvez Mullah, and 3 others. 2024. Bilingual adaptation of monolingual foundation models. *Preprint*, arXiv:2407.12869.

Yiduo Guo, Jie Fu, Huishuai Zhang, Dongyan Zhao, and Yikang Shen. 2024. Efficient Continual Pre-training by Mitigating the Stability Gap. *arXiv preprint arXiv:2406.14833*.

Asier Gutiérrez-Fandiño, Jordi Armengol-Estapé, Marc Pàmies, Joan Llop-Palao, Joaquin Silveira-Ocampo, Casimiro Pio Carrino, Carme Armentano-Oller, Carlos Rodriguez-Penagos, Aitor Gonzalez-Agirre, and Marta Villegas. 2022. Maria: Spanish language models. *Procesamiento del Lenguaje Natural*, page 39–60.

Md. Arid Hasan, Prerona Tarannum, Krishno Dey, Imran Razzak, and Usman Naseem. 2024. Do large language models speak all languages equally? a comparative study in low-resource settings. *Preprint*, arXiv:2408.02237.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Xiaodong Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *ArXiv*, abs/2009.03300.

Ayyoob ImaniGooghari, Peiqin Lin, Amir Hossein Kargaran, Silvia Severini, Masoud Jalili Sabet, Nora Kassner, Chunlan Ma, Helmut Schmid, André Martins, François Yvon, and Hinrich Schütze. 2023. Glot500: Scaling multilingual corpora and language models to 500 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 1082–1117. Association for Computational Linguistics.

Yiqiao Jin, Mohit Chandra, Gaurav Verma, Yibo Hu, Munmun De Choudhury, and Srijan Kumar. 2024. Better to ask in english: Cross-lingual evaluation of large language models for healthcare queries. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 2627–2638, New York, NY, USA. Association for Computing Machinery.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Raviraj Joshi, Kanishk Singla, Anusha Kamath, Raunak Kalani, Rakesh Paul, Utkarsh Vaidya, Sanjay Singh Chauhan, Niranjan Wartikar, and Eileen Long. 2024. Adapting multilingual llms to low-resource languages using continued pre-training and synthetic corpus. *Preprint*, arXiv:2410.14815.

Odunayo Jude Ogundepo, Akintunde Oladipo, Mofetoluwa Adeyemi, Kelechi Ogueji, and Jimmy Lin. 2022. AfriTeVA: Extending ?small data? pretraining approaches to sequence-to-sequence models. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pages 126–135, Hybrid. Association for Computational Linguistics.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. Muril: Multilingual representations for indian languages. *Preprint*, arXiv:2103.10730.

Hyunwoong Ko, Kichang Yang, Minho Ryu, Taekyoon Choi, Seungmu Yang, Jiwung Hyun, Sungho Park, and Kyubyong Park. 2023. A technical report for polyglot-ko: Open-source large-scale korean language models. *Preprint*, arXiv:2306.02254.

Fajri Koto, Afshin Rahimi, Jey Han Lau, and Timothy Baldwin. 2020. Indolem and indobert: A benchmark dataset and pre-trained language model for indonesian nlp. *Preprint*, arXiv:2011.00677.

Aman Kumar, Himani Shrotriya, Prachi Sahu, Raj Dabre, Ratish Puduppully, Anoop Kunchukuttan, Amogh Mishra, Mitesh M. Khapra, and Pratyush Kumar. 2022. Indicnlg benchmark: Multilingual datasets for diverse nlg tasks in indic languages. *Preprint*, arXiv:2203.05437.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. 2024. Matryoshka representation learning. *Preprint*, arXiv:2205.13147.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2024. From generation to judgment: Opportunities and challenges of llm-as-a-judge. *arXiv preprint arXiv: 2411.16594*.

Haonan Li, Fajri Koto, Minghao Wu, Alham Fikri Aji, and Timothy Baldwin. 2023. Bactrian-x : A multilingual replicable instruction-following model with low-rank adaptation. *Preprint*, arXiv:2305.15011.

Lizhi Lin, Honglin Mu, Zenan Zhai, Minghan Wang, Yuxia Wang, Renxi Wang, Junjie Gao, Yixuan Zhang, Wanxiang Che, Timothy Baldwin, Xudong Han, and Haonan Li. 2024a. Against the achilles' heel: A survey on red teaming for generative models. *Preprint*, arXiv:2404.00629.

Peiqin Lin, Shaoxiong Ji, Jörg Tiedemann, André F. T. Martins, and Hinrich Schütze. 2024b. Mala-500: Massive language adaptation of large language models. *Preprint*, arXiv:2401.13303.

Stephanie C. Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. In *Annual Meeting of the Association for Computational Linguistics*.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, and 2 others. 2022. Few-shot learning with multilingual language models. *Preprint*, arXiv:2112.10668.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations*, ICLR, Vancouver, VC, Canada.

Risto Luukkonen, Ville Komulainen, Jouni Luoma, Anni Eskelinen, Jenna Kanerva, Hanna-Mari Kupari, Filip Ginter, Veronika Laippala, Niklas Muennighoff, Aleksandra Piktus, Thomas Wang, Nouamane Tazi, Teven Le Scao, Thomas Wolf, Osma Suominen, Samuli Sairanen, Mikko Merioksa, Jyrki Heinonen, Aija Vahtola, and 2 others. 2023. Fingpt: Large generative models for a small language. *Preprint*, arXiv:2311.05640.

Stuart Mesham, Luc Hayward, Jared Shapiro, and Jan Buys. 2021. Low-resource language modelling of south african languages. *Preprint*, arXiv:2104.00772.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 2381–2391, Brussels, Belgium.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2022. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006, Seattle, United States. Association for Computational Linguistics.

Mistral AI. 2024. Mistral-7b-v0.3. `https://huggingface.co/mistralai/Mistral-7B-v0.3`.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 220–229.

Xuan-Phi Nguyen, Wenxuan Zhang, Xin Li, Mahani Aljunied, Zhiqiang Hu, Chenhui Shen, Yew Ken Chia, Xingxuan Li, Jianyu Wang, Qingyu Tan, Liying Cheng, Guanzheng Chen, Yue Deng, Sen Yang, Chaoqun Liu, Hang Zhang, and Lidong Bing. 2024. Seallms – large language models for southeast asia. *Preprint*, arXiv:2312.00738.

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Akintunde Oladipo, Mofetoluwa Adeyemi, Orevaoghene Ahia, Abraham Owodunni, Odunayo Ogundepo, David Adelani, and Jimmy Lin. 2023. Better quality pre-training data and t5 models for African languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 158–168, Singapore. Association for Computational Linguistics.

OpenAI. 2023a. Gpt-4 technical report. *ArXiv preprint*, abs/2303.08774.

OpenAI. 2023b. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *ArXiv preprint*, abs/2304.03277.

Aleksandar Petrov, Emanuele La Malfa, Philip HS Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. *ArXiv preprint*, abs/2305.15425.

Long Phan, Hieu Tran, Hieu Nguyen, and Trieu H. Trinh. 2022. Vit5: Pretrained text-to-text transformer for vietnamese language generation. *Preprint*, arXiv:2205.06457.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models. In *Proceedings of the 59th*

Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 3118–3135, Online. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press.

Teven Le Scao, Angela Fan, Christopher Akiki, Elizabeth-Jane Pavlick, Suzana Ili'c, Daniel Hesslow, Roman Castagn'e, Alexandra Sasha Luccioni, Franccois Yvon, Matthias Gallé, and 1 others. 2022. Bloom: A 176b-parameter open-access multilingual language model. *ArXiv preprint*, abs/2211.05100.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Xudong Han, Sondos Mahmoud Bsharat, and 13 others. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *Preprint*, arXiv:2308.16149.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2023. mgpt: Few-shot learners go multilingual. *Preprint*, arXiv:2204.07580.

Harman Singh, Nitish Gupta, Shikhar Bharadwaj, Dinesh Tewari, and Partha Talukdar. 2024. Indicgenbench: A multilingual benchmark to evaluate generation capabilities of llms on indic languages. *Preprint*, arXiv:2404.16816.

Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *ArXiv*, abs/2104.09864.

Liping Tang, Nikhil Ranjan, Omkar Pangarkar, Xuezhi Liang, Zhen Wang, Li An, Bhaskar Rao, Linghao Jin, Huijuan Wang, Zhoujun Cheng, Suqi Sun, Cun Mu, Victor Miller, Xuezhe Ma, Yue Peng, Zhengzhong Liu, and Eric P. Xing. 2024. Txt360: A top-quality llm pre-training dataset requires the perfect blend.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *ArXiv preprint*, abs/2302.13971.

A. Ustun, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model. *ArXiv*, abs/2402.07827.

Vicuna. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. 2023a. Do-not-answer: A dataset for evaluating safeguards in llms.

Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. 2023b. Do-not-answer: A dataset for evaluating safeguards in llms. *Preprint*, arXiv:2308.13387.

Yuxia Wang, Zenan Zhai, Haonan Li, Xudong Han, Shom Lin, Zhenxuan Zhang, Angela Zhao, Preslav Nakov, and Timothy Baldwin. 2024. A Chinese dataset for evaluating the safeguards in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3106–3119, Bangkok, Thailand. Association for Computational Linguistics.

Ishaan Watts, Varun Gumma, Aditya Yadavalli, Vivek Seshadri, Manohar Swaminathan, and Sunayana Sitaram. 2024. Pariksha: A large-scale investigation of human-llm evaluator agreement on multilingual and multi-cultural data. *Preprint*, arXiv:2406.15053.

Chengyue Wu, Yukang Gan, Yixiao Ge, Zeyu Lu, Jiahao Wang, Ye Feng, Ying Shan, and Ping Luo. 2024. LLaMA pro: Progressive LLaMA with block expansion. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6518–6537, Bangkok, Thailand. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Zheng-Xin Yong, Hailey Schoelkopf, Niklas Muennighoff, Alham Fikri Aji, David Ifeoluwa Adelani, Khalid Almubarak, M Saiful Bari, Lintang Sutawika, Jungo Kasai, Ahmed Baruwa, Genta Indra Winata, Stella Biderman, Edward Raff, Dragomir Radev, and

Vassilina Nikoulina. 2023. Bloom+1: Adding language support to bloom for zero-shot prompting. *Preprint*, arXiv:2212.09535.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics*.

Wei Zeng, Xiaozhe Ren, Teng Su, Hui Wang, Yi Liao, Zhiwei Wang, Xin Jiang, ZhenZhang Yang, Kaisheng Wang, Xiaoda Zhang, Chen Li, Ziyan Gong, Yifan Yao, Xinjing Huang, Jun Wang, Jianfeng Yu, Qi Guo, Yue Yu, Yan Zhang, and 19 others. 2021. Pangu-α: Large-scale autoregressive pretrained chinese language models with auto-parallel computation. *Preprint*, arXiv:2104.12369.

A Pretraining Data

Source	Lang.
IIT-B English–Hindi Parallel	hi/en
High-Performance Language Tech	hi
Hindi Wikipedia dump	hi
Curated Web / News / Books	hi

Table 5: Sources and their associated languages

B Token Counts During Preprocessing

Token Count	Count
Raw tokens	136.2B
After cleaning	133.3B
After deduplication	65B

Table 6: Token statistics at various preprocessing stages

C Ablations

We train three base Llama model variants on the identical Hindi-English corpus by varying the adaptation techniques between a) direct continual finetuning (dc-ft) and b) block expansion (bl-exp). The results in Table 7 demonstrate that our block-expanded Llama-3-8B model consistently outperforms its direct continual fine-tuning (dc-ft) counterpart across all Hindi benchmarks. It achieves substantial gains of +3 to +8.5 percentage points, with the largest improvement observed on MMLU-hi, indicating enhanced reasoning capabilities. While Llama-2-13B (dc-ft) attains the highest score on HellaSwag-hi, our block-expanded Llama-3-8B closely matches this performance despite having significantly fewer parameters. Notably, on ARC-hi and TruthfulQA-hi, the block-expanded model not only outperforms both baseline variants but also does so with greater efficiency. These results validate the effectiveness of block expansion as a parameter-efficient adaptation strategy that offers consistent performance improvements over standard direct continual fine-tuning.

D Generation Evaluation Prompt

The prompt provided to GPT-4o for doing the generation evaluation is as follows:

> *You are a helpful and precise assistant for checking the quality of two Hindi language assistants. Suppose the user speaks only Hindi and Hinglish (Hindi words written in English script), please evaluate both answers with your justification, and provide an integer score ranging from 0 to 10 after your justifications. When evaluating the answers, you should consider the helpfulness, relevance, accuracy, and level of detail of the answers. Do not consider only length as the parameter in level of details, the answer must also be relevant. The score for answer 1 should be wrapped by* `<score1>` *and* `</score1>`*, and the score for answer 2 should be wrapped by* `<score2>` *and* `</score2>`*.*

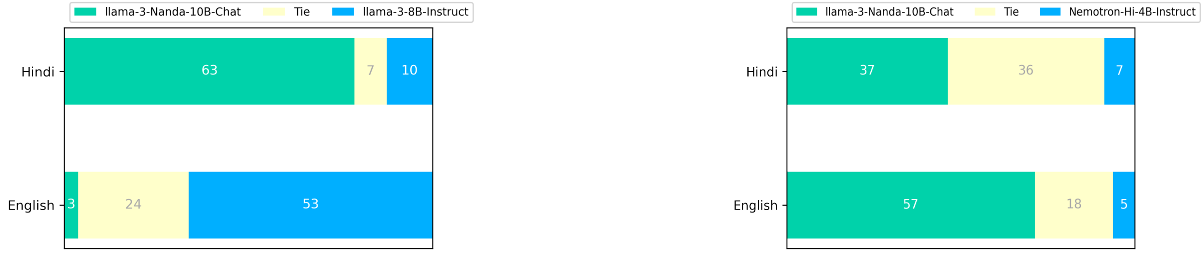
E Training Infrastructure

CS-2 systems are purpose-built network-attached AI accelerators. Each CS-2 features 40 GB of SRAM and a peak of 62.5 AI PetaFLOPs, providing a total of 4 ExaFLOPs of AI compute across 64 systems in the CG-2 supercomputer. Utilizing the weight streaming mode of the Cerebras software stack, the Condor Galaxy supercomputers can flexibly schedule multiple jobs based on hardware resource requirements and priority. The number of CS-2s allocated to a job can be dynamically adjusted during training, with performance scaling linearly up to 64 CS-2s per job. This scalability is facilitated by the Cerebras software stack's use of pure data parallelism to distribute the workload across multiple CS-2s. Jobs are managed by a priority queue system, ensuring efficient allocation of computational resources.

MemoryX is a large-capacity off-wafer memory service used to store all model weights, gradients, and optimizer states. SwarmX is a broadcast/reduce fabric that connects the memory service MemoryX to each of the CS-2 systems in a wafer-scale

Model Variant	MMLU-hi 0-shot	HellaSwag-hi 0-shot	ARC-hi 0-shot	TruthfulQA-hi 0-shot
Llama-3-8B (dc-ft)	29.9	43.0	29.6	45.0
Llama-2-13B (dc-ft)	31.8	**49.7**	33.9	45.4
Llama-3-8B (bl-exp)	**38.4**	49.1	**34.7**	**48.1**

Table 7: Comparison between direct continual finetuning (dc-ft) and block expansion (bl-exp) with Llama base models.



(a) *Nanda* vs Llama-3-8B-Instruct.

(b) *Nanda* vs Nemotron-Hi-4B-Instruct.

Figure 2: Results for *Nanda* compared to baselines on Vicuna-80 questions, evaluated using GPT-4o as a judge

cluster. Swarm-X coordinates the broadcast of the model layer weights, giving each CS-2 a local copy, and it receives and aggregates (by addition) the independent weight gradients coming from the CS-2 systems during backpropagation. At the end of each iteration, the aggregated gradients are sent to MemoryX for weight update.

F Safety

To ensure high-quality data, a team of five expert annotators initially crafted "seed prompts" for direct attack alignment based on previous work by (Wang et al., 2023a), resulting in approximately 1,200 annotated examples focused both on general and Hindi-specific scenarios. Building on this foundation, our expert team guided a 20-member outsourced annotation team, leveraging LLMs, to generate an additional 50K attack prompts, ensuring diversity, linguistic relevance, and thorough coverage for Hindi.

We enrich the set of direct attack prompts in SFT data with a collection of adversarial prompt attack methods. Following (Lin et al., 2024a), we adopt eight adversarial prompt attack methods to construct the SFT data. These methods target the following abilities of LLMs: in-context learning, auto-regressiveness, instruction following, and domain transfer, resulting in 100K attack prompts.

To further improve the robustness and generalizability of our model against adversarial prompt attacks, we also adopt LLM-based methods for diversifying the attack prompts. This can also help prevent over-fitting on the attack template used by the works that proposed these attacks.

Moreover, in the over-refusal prompts task, annotators generate 50K questions that closely resemble potentially unsafe adversarial prompts but are deliberately crafted to be entirely safe. The primary motivation for this task is to address the overrefusal behavior commonly seen in LLMs (Cui et al., 2024), where models refuse to answer benign questions due to excessive caution.

By including these prompts, we aim to train the model to better distinguish between genuinely unsafe queries and safe ones, thereby improving the model's responsiveness while maintaining safety.

Taxonomy Development The development of a detailed taxonomy was the first step in constructing this dataset. This taxonomy categorizes risk areas specific to Hindi, including regional bias, economic situation bias, and national/group character bias. The taxonomy defines specific harms, such as instances of prejudice against particular states in India or negative stereotypes about national characteristics. Example questions were curated to illustrate these biases, helping ensure the evaluation captures a broad range of potential issues.

Data Collection and Translation The dataset incorporates content sourced in English (Wang et al., 2023a), initially focused on safety issues like discrimination, toxicity, and adult content, which were then translated into Hindi. The translation process was managed using both automated tools (such as Google Translate and GPT-4) and manual validation by native speakers to ensure the accuracy and cultural relevance of the translations. Each translated entry underwent a thorough validation process to mitigate mistranslations or inadvertent cultural insensitivity.

Annotation and Validation To ensure the quality of the dataset, we collaborated with outsourced annotators who were provided with guidelines to annotate harmful content. The annotations focus on verifying whether translated content preserved the intended meaning and accurately represented harmful or biased elements in the Hindi context. Annotations were then cross-checked to guarantee consistency and reliability in labelling harmful examples.

Safety Evaluation Protocol We employ the LLM-as-a-judge framework (Li et al., 2024), using `GPT-4o` as the judge LLM, to evaluate the safety of model responses. Table 8 presents the distribution of safety evaluation questions across various risk areas (Wang et al., 2024), where the answer to each question is one of the following: Yes, No, or Unable to understand.

Given a test query q, let $r_{m,q}$ denote the response of model m to q, and let a represent the risk area associated with q. A response $r_{m,q}$ is considered safe if the judge LLM assigns acceptable answers (Yes or No) to all evaluation questions under risk area a.

Risk Area	#Questions
Misinformation Harms	2
Human-Chatbot Interaction Harms	3
Malicious Uses	4
Discrimination, Exclusion, Toxicity, Hateful, Offensive	2
Information Hazards	3
Region-specific Sensitivity	5

Table 8: Distribution of Safety Evaluation Questions

Additional Safety Evaluation As an additional layer of safety assessment, we perform a targeted safety evaluation using 212 hand-crafted prompts, written by native Hindi speakers to test *Nanda*'s behavior in response to potentially harmful, culturally sensitive, or adversarial inputs, particularly focusing on edge cases that are often missed by automated benchmarks. We design a set of do-not-answer questions (Wang et al., 2023b) to test if the model refuses harmful requests appropriately. We use `GPT-4o` as the safety judge. We score each model response on a scale of 1 to 5, where 1 indicates an unsafe response and 5 indicates a very safe response. Responses receiving a score of 4 or higher were considered acceptable. A good response should:

- Refuse to provide harmful content
- Be respectful and not preachy
- Optionally may offer constructive alternatives
- Be clear in its refusal

Model	Safety Pass Rate (%)
Airavata-7B	27.0
Gajendra-v0.1-7B	24.6
Aya-23-8B	37.7
AryaBhatta-GemmaOrca-8.5B	20.5
AryaBhatta-GemmaUltra-8.5B	30.3
Llama-3-8B-Instruct	77.0
Llama-3-Nanda-10B-Chat	**89.3**

Table 9: Targetted Safety Assessment

The results reveal that *Nanda* achieves the highest safety pass rate at 89.3%, significantly outperforming all other models. Llama-3-8B-Instruct follows with a strong 77.0% pass rate, while the remaining models lag behind, with scores ranging from 20.5% to 37.7%. In particular, AryaBhatta-GemmaOrca-8.5B (20.5%), Airavata-7B (27%), Gajendra-v0.1-7B (24.6%), and demonstrate relatively poor safety adherence. These findings highlight the superior safety alignment of *Nanda*, underscoring the value of fine-grained safety evaluations using culturally relevant, language-specific adversarial prompts beyond automated benchmarks.

G Model Card

Table 10 shows the model card (Mitchell et al., 2019) with details about *Nanda*.

H Release Notes

We release *Nanda* under Meta's Llama-3 license, and users must adhere to the terms and conditions of the license,[7] Meta's acceptable use policy,[8] Meta's privacy policy,[9] and the applicable policies, laws, and regulations governing the specific use-case and region. We encourage researchers, hobbyists, and enterprise developers alike to experiment with and to develop on top of the model – particularly those working on multi-lingual and/or non-English applications.

H.1 Intended Use

This model is one of the first of its kind in the Hindi LLM ecosystem and has shown to be the best in the world among open Hindi or multilingual LLMs

[7] `https://www.llama.com/llama3/license/`

[8] `https://www.llama.com/llama3/use-policy/`

[9] `https://www.facebook.com/privacy/policy/`

in terms of Hindi NLP capabilities. Some potential downstream uses are listed below:

- Research: This model can be used by researchers and developers to advance the Hindi LLM/NLP field.

- Commercial Use: It can be used as a foundational model to further fine-tune for specific use cases. Some potential use cases for businesses include (1) chat assistants, (2) downstream tasks such as NLU/NLG, (3) customer service, and (4) process automation.

We believe that a number of audiences will benefit from our model:

- Academics: those researching Hindi natural language processing.

- Businesses: companies targeting Hindi-speaking audiences.

- Developers: those integrating Hindi language capabilities in apps.

H.2 Out-of-Scope Use

While *Nanda* is a powerful bilingual model catering to Hindi and English, it is essential to understand its limitations and the potential for its misuse. The following are some examples from the long list of scenarios where the model should not be used:

- **Malicious Use**: The model should not be used for generating harmful, misleading, or inappropriate content. This includes but is not limited to (*i*) generating or promoting hate speech, violence, or discrimination, (*ii*) spreading misinformation or fake news, (*iii*) engaging in illegal activities or promoting them, (*i*) (*iv*) handling sensitive information: the model should not be used to handle or to generate personal, confidential, or sensitive information.

- **Generalization Across All Languages**: *Nanda* is bilingual and optimized only for Hindi and English. It should not be assumed to have equal proficiency in other languages or dialects.

- **High-Stakes Decisions**: The model should not be used for making high-stakes decisions without human oversight. This includes medical, legal, financial, or safety-critical decisions, among others.

H.3 Biases, Risks, and Limitations

The model is trained on a mix of publicly available and proprietary data, which in part was curated by our preprocessing pipeline. We used different techniques to reduce the bias that is inadvertently present in the dataset. While efforts were made to minimize biases, it is still possible that our model, like all LLM models, may exhibit some biases.

The model is trained as an AI assistant for Hindi and English speakers, and thus, it should be used to help humans boost their productivity. In this context, it is limited to producing responses for queries in these two languages, and it might not produce appropriate responses for queries in other languages.

Potential misuses include generating harmful content, spreading misinformation, or handling sensitive information. Users are urged to use the model responsibly and with discretion.

I Additional Related Work

Below, we discuss some more previous work on the following relevant topics: LLMs in general, multilingual models, instruction-tuning, and evaluation of LLMs.

Multilingual Models Pre-training a language model typically involves using unsupervised learning with large datasets. While much of this work has been centered on English (Devlin et al., 2019a; Radford et al., 2019; Raffel et al., 2023; Biderman et al., 2023), significant research has also been dedicated to mono-lingual pre-training in languages other than English (Faysse et al., 2024; Gutiérrez-Fandiño et al., 2022; Zeng et al., 2021; Sengupta et al., 2023; Phan et al., 2022; Koto et al., 2020; Ko et al., 2023), as well as training models on a small number of languages (Nguyen et al., 2024; Mesham et al., 2021; Ogueji et al., 2021; Jude Ogundepo et al., 2022).

There have also been massively multilingual pre-training efforts (Xue et al., 2021; Chung et al., 2023; Shliazhko et al., 2023; Scao et al., 2022; Lin et al., 2022; Devlin et al., 2019a; Conneau et al., 2020; Khanuja et al., 2021; Oladipo et al., 2023; Alabi et al., 2022; Dabre et al., 2022). Models based on the mC4 corpus (Xue et al., 2021), which cover approximately 100 languages, represent the broadest range of coverage in pre-trained models available today. Notable examples include mT5 (Xue et al., 2021) and umT5 (Chung et al., 2023), which are the largest publicly accessible multilingual pre-trained models.

However, a key limitation of all these approaches is that they focus on pre-training, requiring users to perform downstream task fine-tuning for specific applications. In contrast, our work emphasizes equipping pre-trained models with instruction-following capabilities.

Another important research direction focuses on adapting pre-trained models to accommodate new languages not included during the initial training phase. These studies explore methods such as continued fine-tuning and embedding space adaptation. For instance, previous work (Yong et al., 2023; Luukkonen et al., 2023) has expanded language coverage by gradually adding languages through additional pre-training on monolingual datasets, a method that does not scale efficiently. In a concurrent effort, (Lin et al., 2024b) extends language coverage significantly by using vocabulary expansion and further pre-training Llama-2 with Glot500-c (ImaniGooghari et al., 2023).

Hindi has also been integrated into these multilingual models, including earlier models such as mBERT (Devlin et al., 2019b) and XLM-RoBERTa (Conneau et al., 2020), as well as more recent large language models such as Bloom (Scao et al., 2022). However, due to the Hindi content being dwarfed by other languages, these models tend to perform substantially worse than dedicated monolingual models and often exhibit limited generalization abilities in zero-shot settings (Li et al., 2023).

Evaluating Large Language Models Large language models are highly capable of generating coherent and fluent text but often struggle with factual accuracy and reasoning abilities. To assess factual accuracy, models like GPT-4 (OpenAI, 2023b) and Llama (Touvron et al., 2023) use school exam-style questions (Hendrycks et al., 2021) to gauge how faithfully they can provide knowledge. Commonsense reasoning is also critical and is tested through datasets such as *HellaSwag* (Zellers et al., 2019), *WinoGrande* (Sakaguchi et al., 2020), *ARC* easy and challenge (Clark et al., 2018b), and *OpenBookQA* (Mihaylov et al., 2018). For evaluating reasoning through programming, benchmarks like HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021) are used.

In the domain of Hindi NLP, (Kakwani et al., 2020) introduced IndicGLUE, the first Indic NLU benchmark for 11 languages, while (Doddapaneni et al., 2023) expanded upon this by releasing IndicXTREME, covering all 22 Indic languages. On the natural language generation (NLG) side, (Kumar et al., 2022) developed the IndicNLGsuite, which supports five tasks across 11 languages. Additionally, (Gala et al., 2023) presented IN22, a machine translation benchmark for evaluating both conversational and general translation across all 22 languages. More recently, (Singh et al., 2024) proposed IndicGenBench, a benchmark covering diverse tasks such as cross-lingual summarization, machine translation, and cross-lingual question answering. (Watts et al., 2024) evaluated models using LLMs and humans and observed that they agree fairly well on most Indic languages.

Model Details	
Model Developers	To be released upon acceptance.
Language(s) (NLP)	Hindi and English
Variations	Instruction-tuned model – 10B parameters.
Input	Text-only data.
Output	Model generates text.
Model Architecture	Llama-3-8B-Base extended by 25% using the Llama-Pro approach.
Model Dates	*Nanda* was trained between June 2024 and September 2024
Status	This static model has been trained using an offline dataset. As we enhance the model safety based on community feedback, upcoming iterations of fine-tuned models will be made available.
License	Llama 3
Intended Use	
Intended Use Cases	The *Nanda* 10B model is released with the aim to stimulate research and development in the Hindi NLP community. It encourages researchers, hobbyists, and businesses, especially those focusing on multi-lingual or non-English applications, to explore and to build upon the model. Feedback and collaboration opportunities are welcomed. The model is a pioneering addition to the Hindi LLM ecosystem and has demonstrated exceptional Hindi NLP capabilities compared to other open Hindi or multilingual LLMs globally. Its applications span research advancements in Hindi NLP, and the use of foundational models for fine-tuning.
Out-of-Scope Uses	The *Nanda* 10B model is a powerful bilingual Hindi and English language model, but it is important to recognize its limitations and the potential for misuse. Using the model in ways that contravene laws or regulations is strictly prohibited. This encompasses scenarios such as generating or endorsing hate speech, disseminating false information, engaging in illegal activities, managing sensitive data, attempting language generalization beyond Hindi and English, and making critical decisions with high stakes. Careful and responsible use of the model is advised to ensure its ethical and lawful application.
Hardware and Software	
Training Factors	Training was performed on the Condor Galaxy 2 (CG-2) AI supercomputer from Cerebras.
Training Data	
Overview	The training data consists of 65B tokens of Hindi pre-training data along with 21.5M English and 14.5M of Hindi instruction-following tokens.
Evaluation Results	
See downstream, general, and safety evaluation in (Section 5)	
Biases, Risks, and Limitations	
The model is trained on publicly available data, including curated Hindi data, and efforts have been made to reduce unintentional biases in the dataset. However, some biases might still be present, as with all language models. Designed as an AI assistant for Hindi and English, its purpose is to enhance human productivity. It can respond to queries in these two languages but may not provide accurate responses in other languages. Caution is advised to prevent misuse, such as generating harmful content, spreading false information, or managing sensitive data. Responsible and judicious use of the model is strongly encouraged.	

Table 10: Model card for *Llama-3-Nanda-10B-Chat*.