Domain-adaptative Continual Learning for Low-resource Tasks: Evaluation on Nepali

Sharad Duwal, Suraj Prasai, Suresh Manandhar
Wiseyak
wiseyak.com

Corresponding Author: sharad.duwal@wiseyak.com

Abstract

Continual learning has emerged as an important research direction due to the infeasibility of retraining large language models (LLMs) from scratch in the event of new data availability. Of great interest is the domain-adaptive pre-training (DAPT) paradigm, which focuses on continually training a pre-trained language model to adapt it to a domain it wasn't originally trained on. In this work, we evaluate the feasibility of DAPT in a low-resource setting, namely the Nepali language. We use synthetic data to continue training Llama 3 8B to adapt it to the Nepali language in a 4-bit QLoRA setting. We evaluate the adapted model on its performance, forgetting, and knowledge acquisition. We compare the base model and the final model on their Nepali generation abilities, their performance on popular benchmarks, and run case-studies to probe their linguistic knowledge in Nepali. We see some unsurprising forgetting in the final model, but also surprisingly find that increasing the number of shots during evaluation yields better percent increases in the final model (as high as 19.29% increase) compared to the base model (4.98%), suggesting latent retention. We also explore layer–head self-attention heatmaps to establish dependency resolution abilities of the final model in Nepali. All code will be available at github.com/sharad461/DAPT-Nepali.

1 Introduction

Advancements in natural language processing (NLP) have enabled large language models (LLMs) to generate human-like text, follow instructions and perform well on a wide range of complex understanding tasks (Brown et al., 2020; OpenAI, 2024; Dubey et al., 2024). A big driver behind the continued success of LLMs is the fact that scaling LLMs (increase in parameter count and dataset size) continues to provide decent returns on all performance benchmarks (Kaplan et al., 2020). This scaling-up, however, affects the accessibility and availability of these models and comes with its myriad issues (Bender et al., 2021). Very large language models require huge amounts of resources, have a large carbon footprint (Strubell et al., 2019; Patterson et al., 2021), and training them is feasible only for languages with large quantities of high-quality data and reasonable access to compute. It is costly also to perform inference on them.

Besides scaling, the other direction is generalizability of models with focus on optimal use of data. Given how human text data is projected to run out soon (Villalobos et al., 2024), methods like repeating data, using synthetic data, and using code data are being explored with good returns (Muennighoff et al., 2023; Shimabucoro et al., 2024; Aryabumi et al., 2024). Many of these tools have been explored for research in low-resource languages.

Nepali is a low-resource language. (Arora et al., 2022) classify Nepali among the "Scraping-By" languages in South Asia. While the frontier LLMs today can understand and generate Nepali (OpenAI, 2024), they do not officially support it. One major issue is tokenization: Nepali tokenization is costly in models like GPT4. While NLP research in South Asian languages has picked up recently, many languages are still behind and, as a result, low-resourced.

One possible way to ease the data-compute bind for these low-resource languages (Nepali included) is the use of continual learning (CL) for domain adaptation on high-resource LLMs (SarvamAI, 2023; Gururangan et al., 2020). The idea behind continual learning is to incrementally update an LLM with availability of new data so that the old knowledge isn't forgotten and the new knowledge can be properly assimilated into the model.

Domain adaptation with CL involves continued training of an LLM so that the knowledge of the base LLM can be repurposed to another domain. Since the *knowledge* of the base model can be

Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025), pages 144–153
January 19, 2025.

reused, we do not need large amounts of world knowledge data in the new domain (or language). Also, because adaptation requires training only a fraction of the total parameters in the original model, the compute requirements are significantly reduced.

In this work, we focus on domain adaptation of the Llama 3 8B (Meta, 2024) model to the Nepali language using synthetically generated data. We continually train the Llama model, run experiments to determine performance, catastrophic forgetting, and linguistic knowledge acquisition of the model after the domain adaptation. We compare the adapted model against the original model on several benchmarks. Additionally, we analyze the attention heatmaps to gauge the knowledge of the adapted model. The emphasis of this work is on evaluating DAPT methods to adapt an LLM to a low-resource scenario with only synthetic data.

The main contributions of this work are:

1. We develop and test out methodologies to perform domain-adaptive continual pretraining on an open-weights model using only synthetically generated data.
2. We evaluate and compare the performance of the adapted model against the base model.
3. We interpret the linguistic knowledge of the final model on the new task.

2 Related Work

Continual Learning. Continual learning is an important research direction because its goal is to make it possible to train large models on new data efficiently, often allowing lifelong learning LLMs. This could take place in the form of adding new information to it, teaching it a new subject, or adapting it to a different domain.

Domain-adaptive pretraining (DAPT) has been known to provide performance gains in low-resource settings (Gururangan et al., 2020; Çağatay Yıldız et al., 2024). This has been extended to multilingual domain-adaptive pretraining where a single multilingual model is trained for a specific domain, which outperforms general models on said domain (Kær Jørgensen et al., 2021).

Synthetic data has also been applied for good performance gains in a continual pretraining domain-adaptation strategy (Zhang et al., 2020).

However, a problem in continual learning is catastrophic forgetting, which happens during full finetuning probably due to retraining of weights or because a model has reached knowledge saturation and to learn any more information it forgets old information (Çağatay Yıldız et al., 2024).

Continual learning has great potential in unlocking areas in low-resource language research.

Synthetic data. Data augmentation using synthetic methods is central to research in low-resource languages. In NLP, some methods for synthetic data generation are backtranslation (Sennrich et al., 2016), paraphrasing, synonym replacement, sentence-level replacement, random insertion, etc. (Feng et al., 2021) and (Chen et al., 2023) provide detailed studies on methods available for data augmentation for NLP tasks.

Compared to real data, synthetic data has its own set of advantages and disadvantages. While synthetic data makes low-resource tasks accessible, scalable, and overall cost-effective, it might not always reflect realistic scenarios. There could often be challenges with validating synthetic data and it can magnify biases of the original model.

Organic data available for training purposes is finite and (Villalobos et al., 2024) predict we will run out of all publicly available text data as soon as 2026. Guided synthetic data generation, which will be an important part of future data acquisition technique, is a research direction where data is generated toward non-differentiable objectives (Shimabucoro et al., 2024).

Low-rank adaptation. LoRA (Hu et al., 2021) and QLoRA (Dettmers et al., 2023) are fine-tuning techniques that reduce the number of trainable parameters in a model, making training faster and memory-efficient. Instead of updating all weights in a model, these methods train low-rank matrices that capture task-specific information, freezing the model itself. In addition to the lower rank adaptation in LoRA, QLoRA quantizes the model so that it requires even lesser memory to train. The tradeoff in performance between full finetuning and low-rank techniques has been well-established (Biderman et al., 2024; Xia et al., 2024), and more work is being done in this space (Zhao et al., 2024; Lialin et al., 2023), but in a resource-constrained scenario, QLoRA makes training large models feasible.

Knowledge in attention heads. Many interpretative studies have been applied to the attention mechanism used in Transformer ar-

chitectures. (Voita et al., 2019) investigate the function of attention heads in the multi-head self-attention in encoders and try to interpret how they contribute to the performance of the entire network. They also prune attention heads in an ablation study. Similarly, to analyze how well the attention mechanism models a language and its syntax, (Vig and Belinkov, 2019) evaluate attention heads to find that different layers in a model specialize in different parts-of-speech tags. They use BertViz (Vig, 2019) for their experiments. (Liu et al., 2019) study the contextual representations generated by several popular models to understand why they are so effective in solving NLP tasks. They use seventeen probing tasks to establish the transferability of the representations and what linguistic knowledge is stored and in which part of the model.

3 Method

We use parallel data in Nepali–English (instead of Nepali-only text) to perform continual pretraining. Our aim here is to align the model and its knowledge to Nepali since it already has an understanding of English. We generate the parallel data using synthetic methods, perform pretraining on this data, and then finetune.

3.1 Data Generation

We use Nepali text available online (news reports, essays, etc.) collected in datasets like OSCAR (Abadji et al., 2022) and preprocess it for translation. For the translation system, there were a few alternatives to choose from: NLLB (Costa-jussà et al., 2022), IndicTrans2 (Gala et al., 2023), Google Cloud Translate. We use the Flores test-set for Nepali–English (Guzmán et al., 2019) to evaluate the open-source systems. We also compare scores across the different model sizes available and the various quantized versions of the models. We decided to go with 8-bit NLLB for the translation. IndicTrans2 performed marginally better in terms of BLEU scores, but NLLB had very little computational overhead and supported larger batches out-of-the-box.

Since we also plan to later finetune the model on Nepali instructions and since there aren't instruction sets for Nepali, we also translate English instruction sets to Nepali. We use IndicTrans2 for this. For the instruction set, we translate Alpaca (Taori et al., 2023), Databricks Dolly (Conover et al., 2023) and WebGLM-QA (Liu et al., 2023) to Nepali. To ensure the quality of the synthetic instruction sets, we backtranslate the instructions to English (again using IndicTrans2) and calculate the chrF++ score between the original and the back-translated sets. We apply a chrF++ cut-off of 50 and all samples with lower scores were discarded.

At the end of this step, we have 5M pairs of Nepali–English parallel paragraphs and 114K triplets of (input, instruction, output) instructions.

3.2 Training

We perform 4-bit QLoRA continual pretraining of a Llama 3 8B model on the synthetic parallel data we generated in 3.1. We use Unsloth (Han, 2023). We pretrain the model with the task to translate from English to Nepali. We do this because the English part of the parallel data is synthetic and the Nepali part is organic.

We loosely follow the steps suggested by (SarvamAI, 2023) and divide the pretraining process into two steps:

3.2.1 Pretraining using translation

The aim of this step is to familiarize the model with Nepali using the translation data and the model's own knowledge in English. We train the model to translate from English to Nepali. We use this translation direction because for our parallel data, English is synthetic and Nepali is organic. By training the model to generate the (non-synthetic) Nepali given the (synthetic) English, we teach it to generate Nepali as originally written. The alternative would be to teach the model to generate system-generated English.

For this step, we set the rank to 128, which selected 335M parameters to train. We pretrain the model on 1.5M paragraph pairs for this first task.

3.2.2 Bilingual next token prediction

Second, we train the model on a bilingual next token prediction task. This is the standard next token prediction task with sentences ordered in alternate language. We choose the next 1.5M paragraph pairs and consolidate each of the pairs such that every sample paragraph switches language every sentence. If the first sentence in a paragraph is Nepali, the second picks up in English, then back to Nepali. An example paragraph would be:

> Before the unification of Nepal, the Kathmandu Valley was known as Nepal.
> नेपाल शब्दको सटीक उत्पत्ति अनिश्चित

छ । But it can be dated back to the fourth century AD.

The training settings are much the same for this step as the first step. The presumption here is that instead of training with a Nepali next token prediction task, if we leverage the English knowledge already present into the model, the training should be more effective. (SarvamAI, 2023) found that a model trained with this objective performed better than a model trained on the standard token prediction objective on 5X more data.

3.3 Finetuning

After these steps aligning the model (3.2.1 and 3.2.2) to the Nepali language, we perform a supervised finetuning step. We perform a QLoRA finetuning lower-rank than both these pretraining steps. We set the rank to 16. This updates around 41M parameters in the model. The instruction data we generated in 3.1 is used to finetune the model here. We choose to perform finetuning on a mixed instruction set because we want the model to learn both Nepali and English instructions.

4 Performance Study

After the pretraining followed by finetuning, we perform experiments on both the base model (Llama 3 8B 4-bit) and the continual trained model with the view to answer the following research questions:

Q1. Has the model learned Nepali?
Q2. Has the model retained its knowledge of English? What does catastrophic forgetting look like?
Q3. From a linguistic perspective, how well does the new model model the Nepali language?

5 Experimental Setup

5.1 LM Evaluation Harness

LM Evaluation Harness (Gao et al., 2024) is a framework for evaluating language models. It supports generative LLMs trained on transformers, GPT-NeoX, and Megatron-DeepSpeed and as of writing it supports more than 60 academic benchmarks to run evaluations on. For our task, we focus on English benchmarks and evaluate first the base model, then the adapted model in order to quantify the change in model knowledge and performance.

5.2 BertViz

BertViz (Vig, 2019) is a tool designed to help visualize attention in language models. Originally designed to support only BERT-type models, decoder-only and encoder-decoder model support was added later. It provides a user-friendly interface to explore and interpret the attention patterns within the model, offering valuable insights into how LLMs process and relate different parts of the input with itself or with the output, facilitating in interpretative study of LLMs.

5.2.1 Attention pooling for word tokens

Since Nepali is not officially supported by the Llama 3 tokenizer, the token fertility of Nepali is high. This should be true for many other South Asian languages as well. The study of the attention maps is complicated by this because higher the tokens per word the more difficult it is to map attention between the tokens. Higher fertility not only complicates evaluation, but also makes inference and training costly.

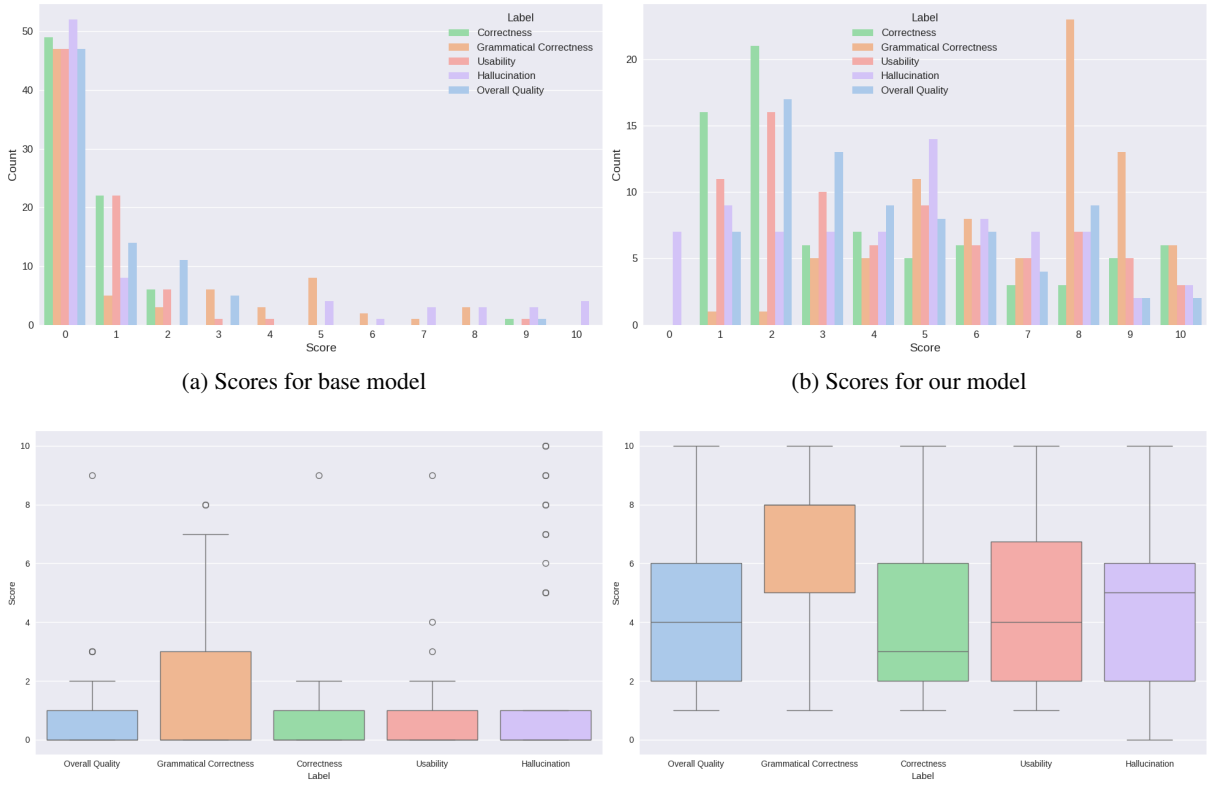
To address this issue, we experimented with methods to pool the token attentions in order to construct word attentions. We applied max-pooling and mean-pooling. For max-pooling, for every Nepali word we take the element-wise max between the vectors corresponding to each constituent token to get the word attention. For mean-pooling, we take the element-wise mean.

Our experiments show max-pooling to be more suitable. We found mean-pooling normalizes attention weights to a great degree, decreasing variance. Thus, for our studies, we max-pool the token attentions to get word attention.

5.3 Questions

For **Q1**, we prompt the base and final models with a set of Nepali questions to generate answers. We then use GPT-4o to score these responses. Automatic evaluation of LM generations has been used with good results due to the multilinguality of frontier language models. GPT4 and GPT-4o perform well even in languages they do not officially support, Nepali included (OpenAI, 2024; Romanou et al., 2024; Hada et al., 2024). We let GPT-4o score the answers on different qualities on scales of 0-10. We analyze the score distributions to answer Q1.

For **Q2**, we use LM Evaluation Harness to evaluate the performance of both the models on several English benchmarks and study how the scores



(a) Scores for base model

(b) Scores for our model

(c) Base model

(d) Our model

Figure 1: GPT4o scores for Nepali answers generated by the base model (Llama 3 8B 4-bit) and our model on five attributes: correctness, grammar, usability, hallucination and overall quality. Empty generations from the models are scored 0 on all attributes. c) and d) are the distribution of scores among the attributes with medians and outliers.

change, or do not. This gives us insight into the forgetting in the final model. Though the base model was trained on eight languages, we only focus on its retention of English-language knowledge. We evaluate the model on MMLU (Hendrycks et al., 2021), ARC (Clark et al., 2018), Winogrande (Sakaguchi et al., 2019), and TruthfulQA (Lin et al., 2022) benchmarks.

Q3. Dependency relations are an important feature of languages. The ability of a language model to resolve a language can be studied by analyzing the layer-head attentions of the model. We use BertViz to analyze the models at the layer- and attention head–level to accomplish this.

We curate Nepali sentences focusing on adjectives and pronouns to study how the layers in the final model encode the information about dependency relation in the sentences. We visualize self-attention in the model.

6 Results

To answer **Q1**, we evaluate text generated by the base model and our model based on five attributes: correctness, grammatical correctness, usability, hallucination tendency, and overall quality.

First, we prompt both the models to answers 78 Nepali questions extracted from a traffic license exam in Nepali. Once we have the generated outputs, we let GPT-4o grade each generation on a scale of 0–10, for all five attributes.

The score distributions in the charts show the distinction between the two models. The base model (Figure 1a) shows a heavy concentration of scores at 0-1 across all metrics. This suggests that the base model's Nepali generation abilities are limited.

Our model has a more balanced score distribution (Figure 1b). While some generations still receive low scores, we observe higher scores overall compared to the base model. This is specifically evident in the scores for grammatical correctness. Our model shows strong performance here, with many generations scoring 8 or above, suggesting the model has learned how Nepali sentence are structured.

Hallucination scores demonstrate that our model has a higher median compared to the base model.

	Our model		Base (Llama 3 8B 4-bit)	
	0-shot	5-shot	0-shot	5-shot
MMLU	**0.3506**	0.3462 (−1.25%)	0.6056	**0.6340** (+4.69%)
ARC-Easy	0.6271	**0.7020** (+11.94%)	0.7950	**0.8346** (+4.98%)
ARC-Challenge	0.3183	**0.3797** (+19.29%)	0.5017	**0.5179** (+3.23%)
Winogrande	0.5801	**0.6275** (+8.17%)	0.7340	**0.7561** (+3.01%)
TruthfulQA MC1	0.2827	-	0.2656	-
TruthfulQA MC2	0.4351	-	0.4305	-

Table 1: Our model v/s the base model on English Benchmarks. As expected, the domain adaptation has caused forgetting. The % change in the scores in 5-shot runs compared to 0-shot runs are also provided. The greater improvements in the 5-shot runs show possible latent retention.

This seems counterintuitive given higher hallucination is a bad quality for a language model to have. But it also suggests that our model's generations contain content that is more verifiable and can be assessed for hallucination, whereas the base model's outputs may be too limited or generic to evaluate factual accuracy.

Both box-plots (Figure 1c and 1d) confirm these observations, evidenced by broader distributions and higher medians for the final model across all metrics.

These results show that our model achieves improvements over the base model across all evaluated dimensions. The broader distribution suggests that our model is capable of generating more sophisticated and varied responses, even though this comes with some increased variability in performance.

For **Q2**, we evaluate the final model on popular English benchmarks in order to identify whether it was able to retain its knowledge in English post-pretraining. The scores of our model versus the base model in the selected benchmarks are reported in Table 1. On MMLU, our model scores 0.3506 and 0.3462 for 0-shot and 5-shot settings respectively. The base model scores 0.6056 and 0.6340 respectively, which suggests some forgetting has taken place.

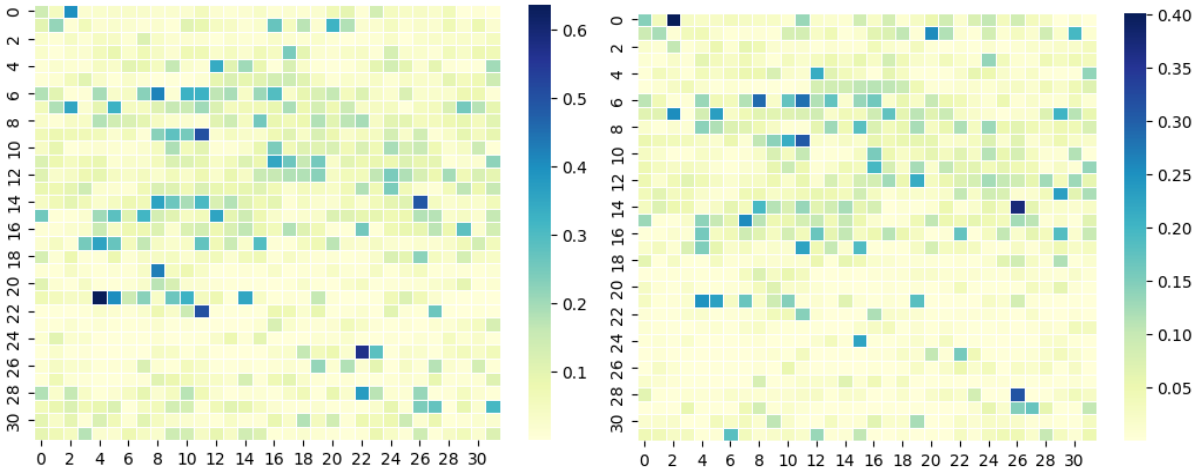
On ARC-Easy, our model achieves scores of 0.6271 (0-shot) and 0.7020 (5-shot), while showing lower performance on the more challenging ARC-Challenge subset with scores of 0.3183 and 0.3797 for 0-shot and 5-shot settings respectively. The base model unsurprisingly scores higher on both benchmarks.

On the Winogrande benchmark, our model scores 0.5691 (0-shot) and 0.6022 (5-shot). For the TruthfulQA evaluation, our model achieves scores of 0.2607 and 0.4243 on MC1 and MC2 variants respectively, showing comparable performance to the baseline's 0.2656 and 0.4305.

With these numbers, it is easy to establish that forgetting has happened. However, it is noteworthy that 5-shot prompting over 0-shot generally yields higher percent increase for our model than the base model, suggesting that our model leverages few-shot examples more effectively than the final model. The highest increase in performance is for the ARC-Challenge dataset where we see a 19.29% performance increase in the 5-shot setting compared to 0-shot. This might suggest that if properly pretrained, forgetting can be curtailed by increasing shots while prompting.

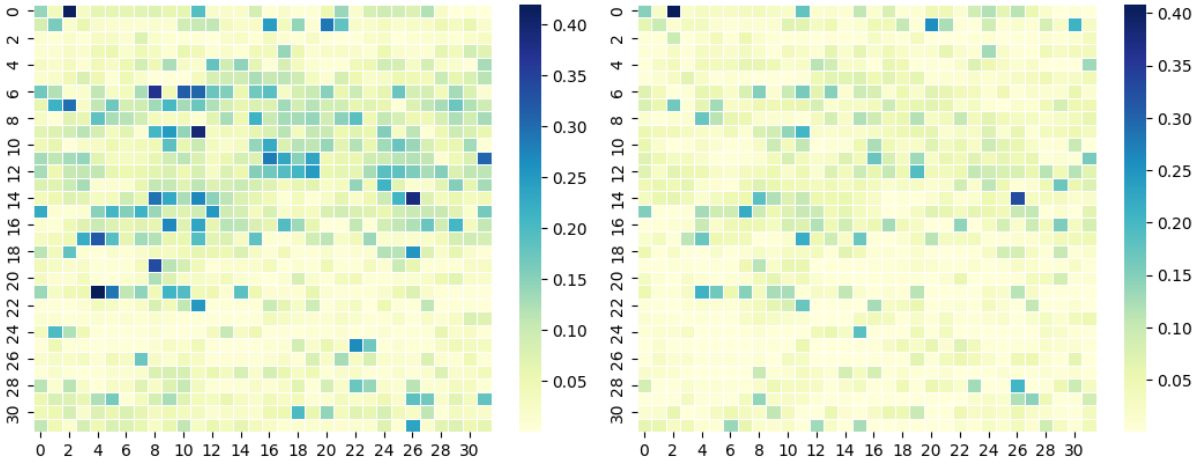
Finally, to answer **Q3**, we annotate a set of Nepali sentences by mapping adjectives to corresponding nouns. We explore the dependency resolution ability of the model by analyzing the attentions from the adjectives to their respective nouns across all attention heads in all layers. For each (adjective, noun) pair we extract attentions across all attention heads and find the mean of such attention heatmaps for multiple samples to get an *adjective concept*. For English we average the heatmaps from 17 adjective-noun pairs and for Nepali 26 pairs. We compare the heatmaps for the base model and the final model to establish whether the final model actually captures some understanding of the language that was not present in the base model. In Figure 2 the heatmaps visualize self-attention patterns across the 32 layers (y-axis) and the 32 attention heads (x-axis) of the models. The darker blue colors indicate stronger attention weights.

Comparing our model's attention heatmaps (a,b) with the base model's heatmaps (c,d), we observe that our model has learned to process Nepali ad-



(a) Our model on Nepali adjectives.

(b) Our model on English adjectives.



(c) Base model on Nepali adjectives.

(d) Base model on English adjectives.

Figure 2: Layer-head heatmaps visualizing attention from adjectives to their respective nouns in Nepali (a,c) and English (b,d) for our model (a,b) and the base model (c,d). Rows are layers and columns are attention heads. From a) and c), we can see our model has learned to attend to Nepali adjectives the way the base model attends to English ones in d).

jectives in a manner very similar to how the base model processes English adjectives. This is evidenced by the sparser and focused attention patterns in (a) as compared to the more diffuse patterns in (c). This alignment suggests improved cross-lingual transfer during pretraining. As suggested in other studies (Liu et al., 2019; Vig and Belinkov, 2019), we found that some of the most prominent attention heads are located in the middle layers.

The models have very different attention patterns in the lower layers (1-8), indicating that language-specific processing is perhaps performed in the earlier layers of the network. The attention patterns for English adjectives (b,d) are similar between the two models, which suggests that the DAPT only impacted the processing of Nepali in the model without disturbing its understanding of English structures.

7 Conclusion

We explored the utility of the continual learning paradigm in low-resource tasks, with a focus on the Nepali language. We experimented with the Llama 3 8B model to establish a simple and intuitive pretraining procedure, followed by mixed-language fine-tuning. We used automatic evaluation to grade model responses and established that the model after DAPT can generate semantically correct Nepali. We performed evaluations with several benchmarks to gauge the forgetting in the model. We finally investigated attention heatmaps

to evaluate the model's grammatical knowledge in Nepali. By adapting a pretrained model to the Nepali language using only synthetic data and very limited resources and establishing generation abilities and linguistic knowledge in the new model, we make a case for domain-adaptive pretraining as a meaningful direction to explore for data- and resource-constrained languages.

8 Limitations

This work focuses on resource-constrained domain adaptation. Experiments are performed in a quantized 4-bit setting and the data used is synthetically generated. Pretraining sessions were run only for a single epoch and the data is mostly from online news sources, which we conjecture lead to more hallucination. Resource constraints are therefore the biggest limitation of this work. Second, we use GPT-4o for evaluation of model output. While auto-evaluation is becoming widely-adopted in multilingual research, use of human evaluators (especially domain experts for Nepali) could lead to a more definitive assessment. Similarly, there are no LM benchmarks in Nepali, which could have helped with the evaluation.

A possible extension of this work could be to study how other low-resourced languages in South Asia respond to these methods. It would also be interesting to investigate if transfer from another Indic language (opposed to English) would yield different results.

References

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. 2022. Towards a Cleaner Document-Oriented Multilingual Crawled Corpus. *arXiv e-prints*, arXiv:2201.06642.

Aryaman Arora, Adam Farris, Samopriya Basu, and Suresh Kolichala. 2022. Computational historical linguistics and language diversity in South Asia. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1396–1409, Dublin, Ireland. Association for Computational Linguistics.

Viraat Aryabumi, Yixuan Su, Raymond Ma, Adrien Morisot, Ivan Zhang, Acyr Locatelli, Marzieh Fadaee, Ahmet Üstün, and Sara Hooker. 2024. To code, or not to code? exploring impact of code in pre-training. *Preprint*, arXiv:2408.10914.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John P. Cunningham. 2024. Lora learns less and forgets less. *Preprint*, arXiv:2405.09673.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2023. An empirical survey of data augmentation for limited data learning in NLP. *Transactions of the Association for Computational Linguistics*, 11:191–211.

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

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, and Loic Barrault et al. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, and Angela Fan et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation

approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

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.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Rishav Hada, Varun Gumma, Adrian de Wynter, Harshita Diddee, Mohamed Ahmed, Monojit Choudhury, Kalika Bali, and Sunayana Sitaram. 2024. Are large language model-based evaluators the solution to scaling up multilingual evaluation? *Preprint*, arXiv:2309.07462.

Daniel Han. 2023. Unsloth ai: Open source fine-tuning for llms.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Rasmus Kær Jørgensen, Mareike Hartmann, Xiang Dai, and Desmond Elliott. 2021. mDAPT: Multilingual domain adaptive pretraining in a single model. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3404–3418, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Vladislav Lialin, Namrata Shivagunde, Sherin Muckatira, and Anna Rumshisky. 2023. Relora: High-rank training through low-rank updates. *Preprint*, arXiv:2307.05695.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. *Preprint*, arXiv:2109.07958.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiao Liu, Hanyu Lai, Hao Yu, Yifan Xu, Aohan Zeng, Zhengxiao Du, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. Webglm: Towards an efficient web-enhanced question answering system with human preferences. *Preprint*, arXiv:2306.07906.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. 2023. Scaling data-constrained language models. *Preprint*, arXiv:2305.16264.

OpenAI. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

David A. Patterson, Joseph Gonzalez, Quoc V. Le, Chen Liang, Lluís-Miquel Munguía, Daniel Rothchild, David R. So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training. *ArXiv*, abs/2104.10350.

Angelika Romanou, Negar Foroutan, Anna Sotnikova, Zeming Chen, Sree Harsha Nelaturu, Shivalika Singh, Rishabh Maheshwary, Micol Altomare, and Mohamed A. Haggag et al. 2024. Include: Evaluating multilingual language understanding with regional knowledge. *Preprint*, arXiv:2411.19799.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *Preprint*, arXiv:1907.10641.

SarvamAI. 2023. Openhathi series: An approach to build bilingual llms frugally.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Luísa Shimabucoro, Sebastian Ruder, Julia Kreutzer, Marzieh Fadaee, and Sara Hooker. 2024. Llm see, llm do: Guiding data generation to target non-differentiable objectives. *Preprint*, arXiv:2407.01490.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. *ArXiv*, abs/1906.02243.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy. Association for Computational Linguistics.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy. Association for Computational Linguistics.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2024. Will we run out of data? limits of llm scaling based on human-generated data. *Preprint*, arXiv:2211.04325.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Wenhan Xia, Chengwei Qin, and Elad Hazan. 2024. Chain of lora: Efficient fine-tuning of language models via residual learning. *Preprint*, arXiv:2401.04151.

Rong Zhang, Revanth Gangi Reddy, Md Arafat Sultan, Vittorio Castelli, Anthony Ferritto, Radu Florian, Efsun Sarioglu Kayi, Salim Roukos, Avi Sil, and Todd Ward. 2020. Multi-stage pre-training for low-resource domain adaptation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5461–5468, Online. Association for Computational Linguistics.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. 2024. Galore: Memory-efficient llm training by gradient low-rank projection. *Preprint*, arXiv:2403.03507.

Çağatay Yıldız, Nishaanth Kanna Ravichandran, Prishruit Punia, Matthias Bethge, and Beyza Ermis. 2024. Investigating continual pretraining in large language models: Insights and implications. *Preprint*, arXiv:2402.17400.