Pretraining Language Models Using Translationese

Meet Doshi[1], **Raj Dabre**[2], **and Pushpak Bhattacharyya**[1]

[1]CFILT, Indian Institute of Technology Bombay, Mumbai, India
[2]National Institute of Information and Communications Technology, Kyoto, Japan
[2]IIT Madras, Chennai, India
[1]{meetdoshi,pb}@cse.iitb.ac.in
[2]raj.dabre@nict.go.jp

Abstract

In this paper, we explore the utility of *Translationese* as synthetic data created using machine translation for pre-training language models (LMs) for low-resource languages (LRLs). Our simple methodology consists of translating large amounts of web-crawled monolingual documents (*clean*) into the LRLs, followed by filtering the translated documents using tiny LMs trained on small but *clean* LRL data. Taking the case of Indian languages, we pre-train LMs from scratch with 28M and 85M parameters, and then fine-tune them for 5 downstream natural language understanding (NLU) and 4 generative (NLG) tasks. We observe that pre-training on *filtered synthetic* data leads to relative performance drops of only 0.87% for NLU and 2.35% for NLG, compared to pre-training on *clean* data, and this gap further diminishes upon the inclusion of a small amount of *clean* data. We also study the impact of *synthetic* data filtering and the choice of source language for *synthetic* data generation. Furthermore, evaluating continually pre-trained larger models like Gemma-2B and Llama-3-8B in few-shot settings, we observe that using *synthetic* data is competitive with using *clean* data. Our findings suggest that *synthetic* data shows promise for bridging the pre-training gap between English and LRLs.

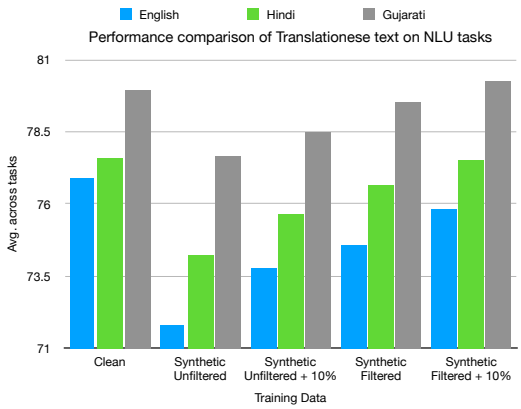

Figure 1: Comparing NLU performance in English, Hindi, and Gujarati shows that filtering synthetic data and adding 10% clean data improves models, approaching the performance of those trained only on clean web data.

1 Introduction

Large language models (LLMs) (Brown et al., 2020; Luccioni et al., 2023; Almazrouei et al., 2023; Lin et al., 2022) have been able to perform very well on downstream tasks like MMLU (Hendrycks et al., 2021), Big-Bench (Srivastava et al., 2022), etc, and have even started to reach human potential in many of these tasks. But this performance has very largely been credited to their scale and the vast amount of data that they have been fed. Most of these language models (LMs) perform well in languages like English where abundant data is available (Kudugunta et al., 2023), but a vast majority of languages don't have comparable data as compared to English. As a consequence, many LLMs, both monolingual and multilingual, involving these languages still show poor performance for various downstream tasks. For example, the largest open source multilingual model BLOOM (Luccioni et al., 2023) covers 46 natural languages spanning 9 language families, but the top 5 languages comprise 74.14% of the data. Even for models like mT5 (Xue et al., 2021), the top 10 of 107 languages account for more than 75.48% of the training data. Despite the benefits of multilingualism (Dabre et al., 2020), this data skew still means that low-resource languages will underperform.

Fortunately synthetic data is an option and previous works such as, but not limited to, back-translation (Sennrich et al., 2016a), sequence distillation (Kim and Rush, 2016), also known as forward translation, etc. have shown that synthetic data obtained using machine translation (MT) can supplement resource scarcity and can significantly enhance model performance (Popović et al., 2020;

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 5843–5862
November 12-16, 2024

Gala et al., 2023). However, to the best of our knowledge, there has been no work on showing the effectiveness of synthetic data for pre-training LMs. Furthermore, the quality of synthetic data is also important, which many works take for granted. While round-trip-translation (Moon et al., 2020) or referenceless neural quality estimation (QE) (Rei et al., 2021) are viable, they either involve twice the compute or a reasonably large model not available for most languages, and this might not be optimal to determine the quality of synthetic documents efficiently. We thus consider TinyLMs (Eldan and Li, 2023) as an efficient alternative, which have been shown to model documents by their fluent paragraph generation capabilities.

In this paper, we focus on Indic languages such as Hindi, Gujarati, and Marathi, and present a comprehensive study of the utility of *synthetic* monolingual data, also called *translationese* (Gellerstam, 1986), obtained using machine translation (MT) for pre-training LMs. We propose a simple framework that involves training tiny language models, henceforth TinyLMs, on original web-crawled data (clean) and then using them to filter synthetic data. We then compare LMs of different scales pre-trained on clean and synthetic data followed by fine-tuning on natural language understanding (NLP) and generation (NLG) downstream tasks, where we observe that, while unfiltered synthetic data based LMs are inferior compared to LMs trained on clean data, filtering leads to performance comparable to the latter. We further show that tuning these synthetic data LMs on small clean data leads to further improvements. We also show that these trends hold when continually pre-training LLMs such as Gemma-2B and Llama-3-8B.

Our contributions are:

a. A simple framework involving high-quality MT models and TinyLMs trained on clean web-crawled data to mass-produce and filter synthetic data for LM training.

b. Demonstrating the efficacy of language models (up to Llama-3-8B) trained on filtered synthetic data across a range of NLU and NLG tasks for low resource Indic languages.

c. A new document-level monolingual corpora (*IndicMonoDoc*) consisting of 39.5B tokens worth of monolingual clean document-level data spanning 22 scheduled languages and English[1].

[1]Our code and datasets are released at https://github.com/meetdoshi90/TranslationesePretraining

2 Related Work

This paper focuses on creating, filtering, and utilizing synthetic data to train TinyLMs.

Monolingual Data: Previous efforts to collect monolingual corpora for Indic languages include the EMILLE/CIIL corpus (McEnery et al., 2000), HindMonoCorp (Bojar et al., 2014), Leipzig corpus (Goldhahn et al., 2012), IndicCorpv1 (Kakwani et al., 2020a), and IndicCorpv2 (Doddapaneni et al., 2023). While IndicCorpv2 is large, it is sentence-level and suitable for NLU models but not for longer contexts needed by language models. We extend these corpora and demonstrate the effectiveness of using synthetic data.

Synthetic Data Generation and Quality Estimation: Synthetic data aids NLP tasks like back translation for machine translation (Sennrich et al., 2016a; Edunov et al., 2018; Marie et al., 2020; Bogoychev and Sennrich, 2019; Ni et al., 2022) and native language identification (Goldin et al., 2018). However, using synthetic data for pre-training LMs is less explored due to hallucination (Maynez et al., 2020) and ungrounded text (Thorne et al., 2018). Evaluation methods like RTT BLEU scores are computationally intensive, while others like BARTScore (Yuan et al., 2021), T5Score (Qin et al., 2023), MQM, and COMET (Rei et al., 2020) require large-scale models or human annotations, limiting scalability. Approaches like KenLM (Heafield, 2011) have been used to filter monolingual corpora based on perplexity.

Transfer Learning and Cross-Lingual Fine-Tuning: Approaches like translate-train, as described by Conneau et al. (2018), involve fine-tuning a multilingual PLM using machine-translated training data and evaluating in the target language. Oh et al. (2022) combined translate-train and translate-test for improved cross-lingual fine-tuning. In contrast, our work focuses on pretraining language models and exploring how synthetic text impacts pretraining and various downstream NLU and NLG tasks.

TinyLMs: Small LMs, even with 10M parameters, produce fluent and consistent text (Eldan and Li, 2023). Challenges like BabyLM (Warstadt et al., 2023) focus on improving LMs within a fixed data budget. We take motivation from this and leverage TinyLMs for efficient filtering of synthetic documents.

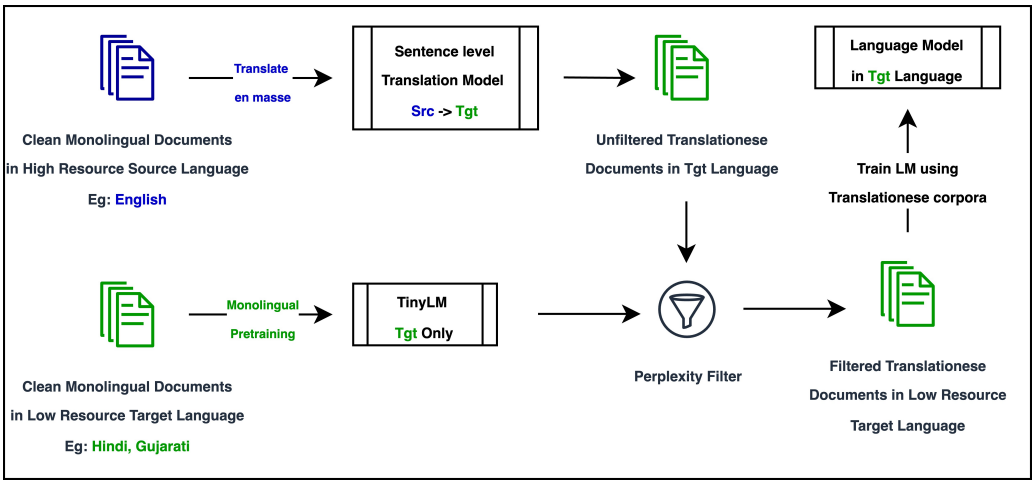

Figure 2: Overview of our approach to pre-train language models using translationese data. We leverage rich monolingual corpora in the *src* language and scarce corpora in the *tgt* language. Our method involves employing a pre-trained machine translation model to translate *src* to *tgt*. We then filter, using perplexity, the resulting text using a TinyLM trained solely on clean *tgt* monolingual data. The filtered synthetic data can be used to further pretrain larger language models.

3 Methodology

In this section, we describe our framework for leveraging synthetic data for LM training. This process consists of collecting monolingual (*clean*) data from the web for low-resource languages, training TinyLMs with it, translating *clean* data from a high resource language such as English into low-resource languages, using the aforementioned TinyLMs to filter *synthetic* data, and then using this filtered data to train LMs for downstream tasks. Our framework is illustrated in Figure 2.

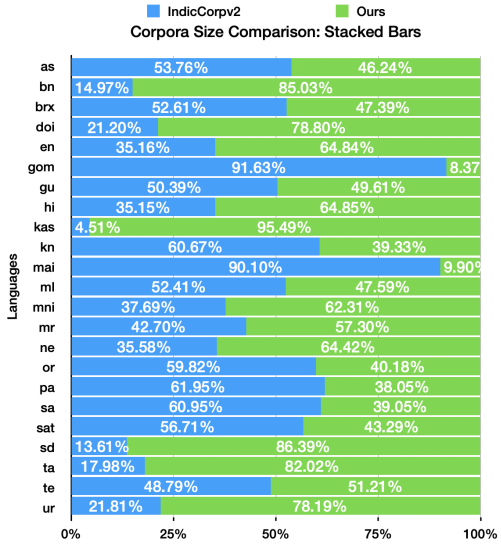

Figure 3: Language-wise corpora size comparison with IndicCorpv2 (Doddapaneni et al., 2023): Stacked Bars

3.1 Collecting Clean Monolingual Corpora

Following Doddapaneni et al. (2023); Rae et al. (2022); Team et al. (2022), for all languages of interest, we **a.** obtain a list of URLs to be crawled via word-level n-grams passed to a search engine, **b.** after URL deduplication, we crawl all applicable webpages, **c.** automatically and manually (Ortiz Suárez et al., 2019; Abadji et al., 2022) filter out unwanted text like HTML tags and emoticons, **d.** use language detection-based (LID) filtering using cld3[2] and IndicLID-FTN model (Madhani et al., 2023a) to discard languages not of interest, **e.** perform document filtering to drop documents containing offensive text using toxic words list provided by Team et al. (2022), **f.** merge all the filtered corpus with Wikipedia, OSCAR (Ortiz Suárez et al., 2019) and some dumps of mC4 (Xue et al., 2021) and finally, **g.** perform deduplication at paragraph level using Murmurhash algorithm[3] with a 128-bit unsigned hash for each monolingual split of the corpora.

We crawl data for English, with Indic context, and 22 Indic languages. As a result, we end up with IndicMonoDoc, with 27.5 billion tokens worth of Indic language documents and 12 billion tokens of English documents for a total of 39.5 billion tokens of *clean* monolingual data. This is larger than the corpora released by Doddapaneni et al. (2023), surpassing it by almost 2 times. We use IndicMon-

[2] https://github.com/google/cld3
[3] https://pypi.org/project/mmh3/

oDoc for all experiments with *clean* data. Figure 3 gives an overview of the comparison of IndicMonoDoc. Note that, creation of IndicMonoDoc is important since IndicCorpV2 is a sentence-level corpus, and training LMs need a document-level corpus. It is important to note that we paid special attention to the low-resource languages. In this paper we only use data corresponding to English, Hindi, Gujarati and Marathi. We report additional details of IndicMonoDoc in Appendix E.

3.2 Generating Translationese (Synthetic)

We utilize state-of-the-art MT models like IndicTrans2 (Gala et al., 2023) to generate translationese data. Using beam search with a beam value of 5, we translate English tokens from the *clean* corpus to the target languages. Due to token limits in MT models, we split documents using the Moses Sentence Splitter[4] for sentence-level translations, then merge them back into documents. We use the 1B En-Indic version[5] of IndicTrans2 to translate 5B English tokens worth of documents from IndicMonoDoc into *translationese* data for Hindi, Marathi and Gujarati.

3.3 Tiny Language Models (TinyLMs)

TinyLMs are small language models inspired by Eldan and Li (2023). We use the Transformer architecture (Vaswani et al., 2017) and train them with *clean* monolingual documents. RoPE embeddings (Su et al., 2023) are used instead of learned positional encodings for handling long documents. Following Chinchilla scaling laws (Hoffmann et al., 2022), we use compute-optimal word tokens. Although it is plausible to train a TinyLM on *unfiltered translationese* data to filter itself, our preliminary experiments revealed that they favor poor-quality data and hence we avoid this route.

3.4 Synthetic Data Filtering

We use these TinyLMs to filter the generated *translationese* data. We do this by using perplexity as a measure of document quality score. For language models, perplexity quantifies how well a model predicts a sequence of tokens. A lower perplexity indicates a natural document. It is calculated by:

$$\mathrm{PPL}(W) = \exp\left\{-\frac{1}{N}\sum_{i}^{N}\log p_\theta\left(w_i \mid w_{<i}\right)\right\}$$

where the negative log-likelihood measures the error of the model's predictions. While calculating perplexity over a sequence of tokens, $W \in w_1, w_2, \ldots, w_N$ we skip the first s tokens where $s = 10$, $e = 1024$ and calculate loss until only the first e tokens of the document. We find setting e to larger values can lead to higher variance in the document scores due to the size of the TinyLM. Following initial analyses, we choose s and e such that we remove the high uncertainty of the language at the start of an unseen document and avoid penalizing longer documents due to the fragility of the extrapolation ability of TinyLM[6]. Note that it is important to choose e such that the language model gives a uniform estimate of perplexity over an already seen sequence of tokens $\in w_s, w_{s+1}, \ldots, w_e$. For our experiments, we use the TinyLMs to score all synthetically generated translationese data and calculate a document score using the above method. Following, Laurençon et al. (2022) we do subsampling by thresholding document perplexity scores except Laurençon et al. (2022) did it using Ken-LM (Heafield, 2011) and we do it using our TinyLM. We keep the threshold value such that we include enough documents to reach the computed optimal token count for pretraining experiments.

4 Experiments

This section outlines the training procedures and datasets for the models described in Section 3. We pre-train decoder only LMs and fine-tune all models from scratch in monolingual and bilingual settings using the causal language modeling (CLM) objective for NLG tasks and a linear classification head for classification tasks. We specify the dataset samples used for pretraining and fine-tuning, and analyze the effects of synthetic corpora on pretraining.

4.1 Pretraining Data Settings

We refer to translated text or translationese as **synthetic** or **syn** and original or web-crawled data as **clean** throughout our experiments. For the pretraining of all base models, we use the following naming convention to denote our training splits for each model:

XX-clean: This is a clean subset sampled randomly from IndicMonoDoc where XX represents the language English (EN), Hindi (HI) or Gujarati (GU).

[4] https://pypi.org/project/mosestokenizer/

[5] https://huggingface.co/ai4bharat/indictrans2-en-indic-1B

[6] During experiments we saw that these TinyLMs can only go up to a certain context length before deteriorating in quality.

syn-XX_yy-unfiltered: Denotes synthetic monolingual documents in XX language generated by using yy as a source during translation.

syn-XX_yy-filtered: Filtered synthetic data.

+10%: Refers to extended pretraining on a cleaned subset of IndicMonoDoc with an additional 10% tokens compared to regular training.

BI-XX-YY Prefix: Denotes bilingual models trained using an equal mixture of monolingual corpora in XX and YY languages. We append an *_syn* prefix to either XX or YY if a synthetic version of that language is employed in training, and a *-parallel/nonparallel* tag to denote whether a parallel version of XX and YY are used or not.

Note, for each split we only use the number of tokens that are required to reach the point of optimality (Hoffmann et al., 2022) by the language model.

4.2 Implementation and Training

Tokenizer: We use a common byte-pair-encoding (BPE) (Sennrich et al., 2016b) tokenizer using Sentencepiece[7] for all experiments. We train a shared vocabulary of 56k subwords between three languages, English, Hindi, and Gujarati by using 5 Million randomly sampled sentences per language and upsampling for Gujarati.

TinyLMs: We use Pytorch Lightning[8] for our implementations and train TinyLMs as described in Section 3.3 for filtering. We use hidden sizes of 768 and have two variants, one with 4 layers (*mini*) and one with 12 layers (*base*; same as GPT2-base) with 28M and 85M non-embedding parameters respectively. The *mini* models are trained on clean data with sequence lengths of 4096[9] (*mini-4k*) for filtering synthetic documents as described in Section 3.4. On the other hand, for our main pre-training and downstream fine-tuning experiments, we train *mini* and *base* models with sequence lengths of 1024 (*mini-1k* and *base-1k*). Following Hoffmann et al. (2022) we use 2.4 billion word tokens per language to compute optimal training of *base* models. Since Gujarati has only 900M tokens in our dataset, whenever Gujarati is involved as the target, we train only the *mini-1k* model. For models involving English and Hindi, we train both *mini* and *base* models. Additional details are in Appendix B.

[7] https://github.com/google/sentencepiece

[8] https://lightning.ai/docs/pytorch/stable/

[9] We keep long sequence lengths to be able to handle long documents for filtering.

4.3 Downstream Tasks and Evaluation

We finetune the *mini-1k* and *base-1k* models for classification, regression, and generation tasks. Hyperparameter tuning is performed using the validation set for models trained only with clean data, and this process is repeated for different data splits. More details on hyperparameters and evaluation can be found in Appendix B. Primary scores are reported on IndicGLUE (Kakwani et al., 2020a) and IndicXNLI (iXNLI) (Aggarwal et al., 2022) for Hindi and Gujarati, and the GLUE benchmark validation set (Wang et al., 2018) for English. We also experiment with other generation tasks like CNN-Dailymail (Nallapati et al., 2016), Dailog-Sum (Chen et al., 2021), XL-Sum (Hasan et al., 2021), IndicNLG (Kumar et al., 2022), FLoRes-200 (Team et al., 2022), IN22-Conv & IN22-Gen (Gala et al., 2023) and use standard evaluation metrics suitable for each task like accuracy, f1-score, Rouge-L (Lin, 2004) and chrF++ (Popović, 2017). Further details about each of the evaluation datasets can be found in Appendix B.1.

5 Results

We now present our results which help establish the utility of synthetic data for language modeling.

5.1 Main Results

In this section, we present results for Hindi, Gujarati, and English language models trained on clean data, as well as synthetic data generated from translations. We demonstrate the impact of filtering and adding additional clean data for extended pretraining of LMs trained solely on synthetic text. Additionally, we observe the effect of using the clean source text along with its translations (synthetic parallel documents) on downstream tasks. We follow the naming convention for different data splits as specified in Section 4. We provide details for the pretraining of each model in Appendix B. We provide additional results in Appendix A.

Filtered Synthetic Data is Competitive with Web Scraped Data: The results in Table 1 and 2 indicate that *syn-HI_en-unfiltered*, *syn-GU_en-unfiltered*, and *syn-EN_hi-unfiltered* exhibit lower downstream performance compared to their filtered counterparts: *syn-HI_en-filtered*, *syn-GU_en-filtered*, and *syn-EN_hi-filtered*, respectively. It is evident that filtering the synthetic documents using TinyLMs significantly improves the performance of both NLU and NLG tasks. We also observe

(a) Results on Hindi

Model	NLU						NLG				
	iXNLI	bbc-a	iitp-mr	iitp-pr	midas	Avg.	Headline Gen.	Sentence Summ.	Question Gen.	Wikibio	Avg.
HI-clean	73.61	81.75	72.58	79.73	80.34	77.60	27.54	23.64	24.84	52.16	32.04
syn-HI_en-unfiltered	72.87	77.92	64.36	76.22	79.91	74.26	27.29	22.93	24.22	50.14	31.14
syn-HI_en-unfiltered+10%	74.63	78.36	67.75	77.46	80.17	75.67	26.98	23.20	24.76	**51.34**	31.57
syn-HI_en-filtered	**74.75**	**81.06**	69.03	78.58	79.73	76.63	27.15	23.10	24.41	49.88	31.13
syn-HI_en-filtered+10%	74.49	80.94	**71.61**	**79.92**	**80.64**	**77.52**	**27.87**	**24.23**	**24.87**	51.18	**32.04**

(b) Results on Gujarati

Model	NLU			NLG			
	iXNLI	iNLTK	Avg.	Headline Gen.	Sentence Summ.	Question Gen.	Avg.
GU-clean	67.8	92.1	79.95	17.62	13.82	15.18	15.54
syn-GU_en-unfiltered	65.51	89.78	77.65	16.21	13.29	13.66	14.39
syn-GU_en-unfiltered+10%	66.83	90.11	78.47	17.28	13.27	14.50	15.02
syn-GU_en-filtered	67.74	91.35	79.55	**17.64**	**13.40**	14.95	**15.33**
syn-GU_en-filtered+10%	**68.04**	**92.41**	**80.23**	17.62	13.16	**15.00**	15.26

Table 1: Results for Hindi and Gujarati: NLU/NLG tasks on *base-1k* (Hindi) and *mini-1k* (Gujarati) models on different clean and synthetic splits. Test accuracy for NLU tasks; Rouge-L F1 scores for NLG tasks. **Bold** values represent the best amongst synthetic splits.

that for tasks like CoLA (Warstadt et al., 2019), language models trained solely on synthetic data lag behind when compared to other tasks as seen in Table 8 of Appendix A. This suggests that *synthetic corpora may lack certain important elements necessary for language models* to perform competitively in linguistic acceptability tasks, as opposed to LMs trained on clean, non-synthetic corpora. Results for *base-1k* for English are presented in Table 8 in Appendix A because we focus our attention on Indic languages.

Fine-tuning on small amounts of Web Scraped Data boosts performance: Even after filtering, we observe that language models trained solely on synthetic text slightly underperform LMs trained on clean text. To address this issue, we conduct extended pretraining of LMs using clean data sourced from IndicMonoDoc. The objective is to determine if this additional training improves performance. We only incorporate an additional 10% of clean data compared to the LM's previous training data. We see these results across all three languages, and for Hindi and Gujarati, we see that *by incorporating even a small amount of clean data, we observe an increase in performance on downstream tasks*, bringing the LM at par or closer to the performance of a clean LM. We see an improvement in LMs trained using unfiltered synthetic corpora as well but we believe that filtering leads to the removal of noisy documents and thus better performance. We observe improved performance in language models (LMs) trained with unfiltered synthetic corpora, but filtering out noisy documents enhances performance further. Our ablation study (Table 13 in Appendix A) investigates whether adding 10% more synthetic data contributes to this improvement or if the data type is key. While performance gains could stem from statistical variances, the consistency across nearly all downstream tasks suggests otherwise.

Impact of source language for synthetic data generation: Choosing the right source language for synthetic corpora is crucial, as it influences the characteristics of the generated translationese text. We evaluate this using Hindi and Gujarati clean documents from *IndicMonoDoc*, translating them into English. We use the 1B Indic-En version[10] to translate 5B Hindi tokens and 900M[11] Gujarati tokens into English. In Table 2, we see that the *synthetic text generated from Hindi achieves performance at par with the EN-clean model, while the synthetic text from Gujarati significantly lags behind*. This is likely because Hindi is more *macaronic* than Gujarati, i.e., a lot of Hindi text from the web consists of *Hinglish*, resulting in better translationese text due to increased overlap between languages. This can also be due to the weaker translations generated by the MT model. The performance gap is notable in tasks like STS benchmark, NLI (qnli and mnli), and CoLA, suggesting

[10] https://huggingface.co/ai4bharat/indictrans2-indic-en-1B

[11] Since Gujarati has limited data (900M tokens), we train a *mini-1k* model for fair comparison.

Model		sst2 acc	cola mcc	mrpc f1	qnli acc	qqp f1	rte acc	mnli-m acc	mnli-mm acc	stsb pearson	Avg.
Original	EN-clean	87.95	25.59	83.84	78.83	80.78	64.62	71.6	71.69	73.48	70.93
Translationese Hi→En	syn-EN_hi-unfiltered	87.53	19.77	79.02	76.49	77.96	55.4	69.65	70.14	67.37	67.04
	syn-EN_hi-filtered	87.61	22.81	81.95	77.63	**80.57**	56.31	70.19	70.89	69.29	68.58
	syn-EN_hi-filtered + 10%	**87.84**	**26.61**	**83.27**	**78.5**	80.36	**61.37**	**71.29**	**71.11**	**71.91**	**70.25**
Translationese Gu→En	syn-EN_gu-unfiltered	83.11	17.66	78.53	66.01	77.68	53.6	63.21	64.55	27.33	59.08
	syn-EN_gu-filtered	85.66	21.15	81.45	66.35	77.36	54.15	66.27	65.72	26.16	60.47
	syn-EN_gu-filtered + 10%	**86.58**	**25.17**	**81.67**	**67.1**	**77.75**	**57.76**	**68.78**	**68.56**	**27.54**	**62.32**

Table 2: Effect of source selection for generating synthetic data on the dev set of GLUE benchmark. All the results reported here are on *mini-1k*. **Bold** values represent the best amongst synthetic splits

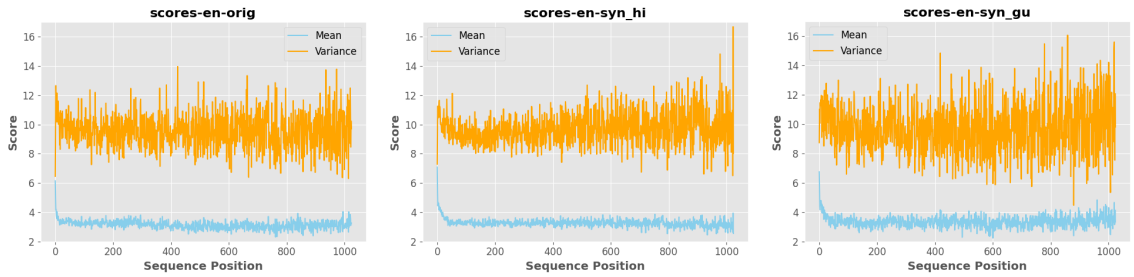

Figure 4: The plot illustrates TinyLM's perplexity mean and variance across various datasets: Clean-EN (left), Syn-EN from filtered Hindi (middle), and Syn-EN from filtered Gujarati (right). Despite filtering, English documents generated from translating Gujarati show consistently higher variance.

poorer translation quality from Gu→ En compared to Hi→ En.

Model	iXNLI	bbc-a	iitp-mr	iitp-pr	midas	Avg.
HI-clean	68.74	80.25	67.74	77.05	78.33	74.42
syn-HI_en-unfiltered	67.32	77.92	65.63	76.81	77.58	73.05
syn-HI_en-filtered	69.48	78.98	65.16	77.43	77.33	73.68
syn-HI_en-filtered+10%	**70.15**	**79.56**	**67.09**	**78.2**	**79.03**	**74.81**

Table 3: Effect of reducing model size for Hindi on IndicGLUE accuracy. All the results reported here are on *mini-1k*. **Bold** values represent the best amongst synthetic splits

5.2 Further Exploration

Analysis of Synthetic Data: Figure 4 shows the perplexity mean and variance scores for TinyLM across token positions in the documents. This shows that on unseen documents, TinyLM shows higher variance on English documents generated by translating Gujarati documents from IndicMonoDoc as compared to English clean and English synthetic generated from Hindi. This also gives reason for the deterioration in results in Table 2 due to Gujarati documents. Figure 6 shows the distribution of lengths of filtered documents by TinyLMs showing that they do not add any bias for shorter documents during filtering.

Impact of model size: Following Table 2 and 3, we see that even after scaling down we see consistent improvements for filtering and adding additional data, which empirically shows that indeed *using synthetic text after filtering is a viable option for pretraining LMs* of varying sizes. In Table 3 we see that after filtering and extended pretraining, synthetic text outperforms LMs trained on clean documents from the web in Hindi. This is also supported by our experiments on finetuning Llama-3-8B in Section 5.3.

Model	XLSum HG	XLSum QG	Cnn	Dialogsum	Avg.
EN-clean	23.87	24.05	16.08	20.39	21.10
syn-EN_hi-unfiltered	22.17	22.97	12.56	18.30	19.00
syn-EN_hi-filtered	**23.27**	**23.83**	**15.88**	**19.83**	**20.70**

Table 4: Performance of English models on NLG tasks. All the results reported here are on *base-1k* and use Rouge-L F1 scores.

Impact on NLG: Without extended pretraining, language models trained on synthetic text perform as well as those trained on clean documents, suggesting that *for NLG tasks, synthetic data suffices for pretraining*, eliminating the need for clean data. This trend is evident across Hindi, Gujarati, and English NLG results (Tables 1 and 4). As their performance matches models trained on clean data, we refrain from extended pretraining for NLG tasks, focusing primarily on abstractive summarization for evaluating generation capabilities.

5.3 Scaling to Llama-3-8B

To show the effect of using translationese on larger models, we select Llama-3-8B[12] and Gemma-2B (Team et al., 2024) and perform continual pretraining over clean and synthetic data to improve ability over the low resource target language. We take Marathi as a replacement for Hindi for scaling experiments since data for Hindi is abundantly available and existing models already have a good language understanding of Hindi making it harder to compare the effects of utilizing clean vs synthetic data. For a fair comparison, we limit each data split to 344M tokens for Gujarati and 465M tokens for Marathi and follow a similar procedure as described in Section 3.4 to generate and filter data for Marathi. We perform extended training for a single epoch using LoRA (Hu et al., 2021) finetuning on W_q, W_v projection matrices using α=16 and r=8. We keep the learning rate at $3e^{-5}$ with a weight decay of 0.01 and an effective batch size of 58k.

Perplexity: We report average sentence level perplexity on sentences from IN22-Conv and IN22-Gen (Gala et al., 2023) in Table 5. We see that filtered synthetic data for Gujarati outperforms clean data, but for Marathi, it does not. This means that *filtering improves performance at scale but relies on the quality of translationese* in the target language. We report perplexities on individual test sets in Appendix A.

Data	Marathi		Gujarati	
	Gemma 2B	Llama-3 8B	Gemma 2B	Llama-3 8B
Base model	178.898	66.740	71.136	2.839
clean	37.599	11.196	10.350	2.312
synthetic-unfiltered	**92.813**	15.697	10.941	2.816
synthetic-filtered	104.148	**14.622**	**10.150**	**2.236**

Table 5: Average perplexity (↓) of models trained on Translationese vs. Clean data on IN22-Conv and IN22-Gen test sets shows improvement with large-scale models. **Bold** represents best among synthetic data splits.

Few Shot Prompting: We evaluate our continually pre-trained models using few-shot prompting on IndicSentiment classification (Doddapaneni et al., 2023) as the NLU task and En→Indic machine translation on IN22-Gen and FloRes-200 as the NLG task. Prompts used are shown in Appendix B.5 with examples are randomly sampled from the validation set for FloRes and other examples from the IN22-Gen test set, ensuring no example is repeated in the prompts. We use a beam width of 5 with early stopping enabled. Results are shown in Tables 6 and 7. We see that filtering improves performance on MT when compared to synthetic splits, but IndicSentiment, shows only marginal improvements. Nonetheless, models trained on filtered data show lower perplexity and better performance in few-shot settings, indicating their promise. We leave the exploration of training multilingual LLMs on large-scale translationese data for future research.

[12] https://github.com/meta-llama/llama3

Data	Flores-200		IN22-Gen	
	Marathi	Gujarati	Marathi	Gujarati
Base model	27.83	34.35	30	34.92
clean	34.02	35.61	33.94	35.42
synthetic-unfiltered	30.63	34.19	29.67	32.1
synthetic-filtered	**31.81**	**35.54**	**31.3**	**35**

Table 6: chrF++ scores on 5-shot Machine Translation FloRes and IN22-Gen test sets on Llama-3-8B. **Bold** represents best among synthetic data experiments.

Data	Marathi		Gujarati	
	Gemma 2B	Llama-3 8B	Gemma 2B	Llama-3 8B
Base model	90.89 $^{\pm 0.005}$	95 $^{\pm 0.009}$	83.84 $^{\pm 0.002}$	92.69 $^{\pm 0.018}$
clean	90 $^{\pm 0.014}$	97.17 $^{\pm 0.013}$	87.79 $^{\pm 0.012}$	93.33 $^{\pm 0.0132}$
synthetic unfiltered	**89.66** $^{\pm 0.012}$	95.38 $^{\pm 0.016}$	83.88 $^{\pm 0.009}$	92.81 $^{\pm 0.01}$
synthetic filtered	86.67 $^{\pm 0.016}$	**96.15** $^{\pm 0.011}$	**84.10** $^{\pm 0.013}$	**92.94** $^{\pm 0.007}$

Table 7: Average accuracy with standard deviation (superscript) over 5 runs on 10-shot IndicSentiment classification task. **Bold** represents the best among synthetic data experiments.

6 Conclusion

In this paper, we performed a first of its kind study showing the promise of using translationese data for training language models for low-resource languages. Our simple pipeline involves the translation of high-resource source language documents at scale, followed by perplexity based filtering using small and efficient language models trained on clean target low-resource language data. We then showed on a variety of downstream natural language understanding and generative tasks that both small and large language models pre-trained on clean synthetic data are comparable to those trained on clean data. While we observed that the source language for synthetic data generation matters, it is clear that synthetic data can help bridge the resource scarcity faced by a vast majority of languages for language modeling. Future work will focus on better and faster synthetic data generation and filtering mechanisms.

Acknowledgements

We extend our thanks to the Department of Computer Science and Engineering at the Indian Institute of Technology (IIT) Bombay for providing access to GPU servers, and to the Centre for Development of Advanced Computing (C-DAC) for granting us access to the Param Siddhi Supercomputer. These computational resources were essential for the successful completion of this work.

Limitations

We consider the following limitations of our work.

- We show that synthetic data also helps for larger models like Llama-3-8B but for even larger models above 100B parameters, effects of translationese may be different. However, synthetic data generated from translations can surely help fill knowledge gaps.
- Due to the extensive size of the test sets for IndicNLG tasks (Question Generation, WikiBio generation, Headline Generation, and Sentence Summarization), we couldn't experiment with them in their entirety. However, since we already use 4000 examples per language, we anticipate that the overall trends remain unchanged.
- We report GLUE validation set results for all models due to the large scale of our experiments, following existing practices. Our goal is to demonstrate synthetic data utility, not to achieve state-of-the-art results.
- Our framework heavily relies on the translation model's performance. Despite this dependency, we are confident that our approach will significantly enhance the performance of mid-resource languages, especially where the translation model is already proficient.

Ethical Considerations

As a part of this paper, we release monolingual and synthetic data. While we have taken care to remove any toxic content, accidental occurrences may exist and thus we exercise caution when using our data for training language models as they may produce toxic outputs. Given that we have shown the utility of synthetic data for training LMs, it should be possible to mass produce synthetic toxic data in various languages leading to LMs that can generate multilingual toxic content. However, this opens up research opportunities on how to detect and filter toxic content from synthetically created data.

We release the code and models with an MIT License[13]. The dataset is released under a CC-0 License[14].

References

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

Divyanshu Aggarwal, Vivek Gupta, and Anoop Kunchukuttan. 2022. IndicXNLI: Evaluating multilingual inference for Indian languages. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10994–11006, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models.

Nikolay Bogoychev and Rico Sennrich. 2019. Domain, translationese and noise in synthetic data for neural machine translation. *CoRR*, abs/1911.03362.

Ondrej Bojar, Vojtech Diatka, Pavel Rychlỳ, Pavel Stranák, Vít Suchomel, Ales Tamchyna, and Daniel Zeman. 2014. Hindencorp-hindi-english and hindi-only corpus for machine translation. In *LREC*, pages 3550–3555.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*,

[13] https://opensource.org/license/mit/

[14] https://creativecommons.org/share-your-work/public-domain/cc0/

pages 5062–5074, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Comput. Surv.*, 53(5).

Daniel Deutsch and Dan Roth. 2020. SacreROUGE: An open-source library for using and developing summarization evaluation metrics. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 120–125, Online. 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. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12402–12426.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

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

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

Martin Gellerstam. 1986. Translationese in swedish novels translated from english. *Translation studies in Scandinavia*, 1:88–95.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).

Gili Goldin, Ella Rabinovich, and Shuly Wintner. 2018. Native language identification with user generated content. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3591–3601, Brussels, Belgium. Association for Computational Linguistics.

Yvette Graham, Barry Haddow, and Philipp Koehn. 2019. Translationese in machine translation evaluation. *CoRR*, abs/1906.09833.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

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.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models.

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

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. 2020a. 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.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020b. 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.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Christopher A Choquette-Choo, Katherine Lee, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, et al. 2023. Madlad-400: A multilingual and document-level large audited dataset. *arXiv preprint arXiv:2309.04662*.

Aman Kumar, Himani Shrotriya, Prachi Sahu, Amogh Mishra, Raj Dabre, Ratish Puduppully, Anoop Kunchukuttan, Mitesh M. Khapra, and Pratyush Kumar. 2022. IndicNLG benchmark: Multilingual datasets for diverse NLG tasks in Indic languages. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5363–5394, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, et al. 2022. The bigscience roots corpus: A 1.6 tb composite multilingual dataset. *Advances in Neural Information Processing Systems*, 35:31809–31826.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. 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, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. 2023. Estimating the carbon footprint of bloom, a 176b parameter language model. *J. Mach. Learn. Res.*, 24:253:1–253:15.

Yash Madhani, Mitesh M. Khapra, and Anoop Kunchukuttan. 2023a. Bhasha-abhijnaanam: Native-script and romanized language identification for 22 indic languages.

Yash Madhani, Sushane Parthan, Priyanka Bedekar, Gokul Nc, Ruchi Khapra, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Khapra. 2023b. Aksharantar: Open Indic-language transliteration datasets and models for the next billion users. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 40–57, Singapore. Association for Computational Linguistics.

Benjamin Marie, Raphael Rubino, and Atsushi Fujita. 2020. Tagged back-translation revisited: Why does it really work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5990–5997, Online. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Anthony McEnery, Paul Baker, Robert Gaizauskas, and Hamish Cunningham. 2000. Emille: Building a corpus of south asian languages. In *Proceedings of the International Conference on Machine Translation and Multilingual Applications in the new Millennium: MT 2000*.

Jihyung Moon, Hyunchang Cho, and Eunjeong L. Park. 2020. Revisiting round-trip translation for quality estimation. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 91–104, Lisboa, Portugal. European Association for Machine Translation.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Jingwei Ni, Zhijing Jin, Markus Freitag, Mrinmaya Sachan, and Bernhard Schölkopf. 2022. Original or translated? a causal analysis of the impact of translationese on machine translation performance. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5303–5320, Seattle, United States. Association for Computational Linguistics.

Jaehoon Oh, Jongwoo Ko, and Se-Young Yun. 2022. Synergy with translation artifacts for training and inference in multilingual tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6747–6754, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pages 9 – 16, Mannheim. Leibniz-Institut für Deutsche Sprache.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Maja Popović, Alberto Poncelas, Marija Brkic, and Andy Way. 2020. Neural machine translation for translating into Croatian and Serbian. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 102–113, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL).

Yiwei Qin, Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2023. T5Score: Discriminative fine-tuning of generative evaluation metrics. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15185–15202, Singapore. Association for Computational Linguistics.

Ella Rabinovich and Shuly Wintner. 2015. Unsupervised identification of translationese. *Transactions of the Association for Computational Linguistics*, 3:419–432.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2022. Scaling language models: Methods, analysis & insights from training gopher.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2022. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

Ricardo Rei, Ana C Farinha, Chrysoula Zerva, Daan van Stigt, Craig Stewart, Pedro Ramos, Taisiya Glushkova, André F. T. Martins, and Alon Lavie. 2021. Are references really needed? unbabel-IST 2021 submission for the metrics shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1030–1040, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. 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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek,

Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Santilli, Andreas Stuhlmüller, Andrew M. Dai, Andrew La, Andrew K. Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakas, and et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *CoRR*, abs/2206.04615.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2023. Roformer: Enhanced transformer with rotary position embedding.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology.

NLLB Team, 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, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The fact extraction and VERification (FEVER) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Antonio Toral, Sheila Castilho, Ke Hu, and Andy Way. 2018. Attaining the unattainable? reassessing claims of human parity in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 113–123, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

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.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing*

Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pages 27263–27277.

Mike Zhang and Antonio Toral. 2019. The effect of translationese in machine translation test sets. *CoRR*, abs/1906.08069.

A Additional results

We report additional results in this section. Tables 14, 15 show the chrF++ and BLEU scores across three translation evaluation benchmarks. This shows that using parallel synthetic data does not deteriorate the performance of the language model. Similar results are shown in Table 16 for IndicNLG tasks where performance on Hindi generation tasks are only affected by a small margin and coupled with results in Table 8 showing that scores are not affected by using Hindi synthetic parallel data.

Using synthetic for one language doesn't impact performance in another: For many multilingual language models, data imbalance causes a gap in performance across languages. But what if we can combine synthetic data along with clean data for training multilingual models? would the synthetic part deteriorate the performance of the multilingual model? To experiment with this, we train bilingual *base-1k* models over different combinations of clean and synthetic corpora for English and Hindi and evaluate their performance on GLUE (Wang et al., 2018), and report performance on IndicNLG, and Machine translation in Appendix A. Following Table 9, we see that using Hindi synthetic data does not affect its performance compared to *BI-EN-HI-clean* model which is solely trained on clean corpora. This implies that *it is possible to train multilingual models where some languages are trained only over a clean subset and others on synthetic* without deteriorating performance across languages. We further see that *using parallel data does not have much impact on multilingual models*.

Impact on Machine Translation: (MT) We focus on MT separately as a special case of NLG. We hypothesized that using parallel synthetic documents for bilingual models would improve translation performance by enhancing alignment between languages. However, our evaluation fails this hypothesis. Results indicate that *using nonparallel synthetic documents yields similar translation performance across language directions and benchmarks compared to parallel synthetic documents*. This might be because there is no explicit alignment happening during training between parallel documents. See Table 10 for chrF++ scores on FLoRes-200 (Team et al., 2022), and Appendix A for chrF++ and BLEU scores on IN22-Conv, IN22-Gen (Gala et al., 2023).

B Training and Evaluation

In this section, we provide an overview of the training and evaluation setup employed in our experiments. This includes details about the datasets used, training hyperparameters, evaluation metrics, and other relevant configurations.

B.1 Evaluation Datasets

For evaluation, we utilize a diverse set of datasets covering four languages: English, Hindi, Gujarati, and Marathi. For Hindi and Gujarati, we rely on the IndicGLUE benchmark[15] (Kakwani et al., 2020b), which provides a range of tasks for natural language understanding (NLU), including natural language inference (IndicXNLI/iXNLI), article genre classification (bbc-a, iNLTK), discourse mode classification (MIDAS), and sentiment analysis (iitp-mr, iitp-pr). For natural language generation (NLG), we employ the IndicNLG benchmark[16] (Kumar et al., 2022), which includes tasks like headline generation, sentence summarization, question generation, and Wikipedia biography generation.

The IndicGLUE dataset is semi-automatically curated using website metadata and Wikipedia articles, while IndicNLG is derived from Wikipedia articles and news websites for summarization, along with parallel corpora and pivot-based translation for paraphrasing tasks. Additionally, we incorporate the test sets from IN22 (Gala et al., 2023) and Flores-200 (Team et al., 2022) to evaluate performance on machine translation tasks.

We use the well-known GLUE benchmark, which includes nine NLU tasks in English such as natural language inference (NLI), semantic similarity, text classification, and linguistic acceptability. For English summarization tasks, we rely on XL-Sum (Hasan et al., 2021), DialogSum (Chen et al., 2021), and CNN/DailyMail (See et al., 2017).

[15] https://huggingface.co/datasets/ai4bharat/indic_glue

[16] https://huggingface.co/collections/ai4bharat/indicnlg-66c5a1397bab135be074cfe1

Model	sst2 acc	cola mcc	mrpc f1	qnli acc	qqp f1	rte acc	mnli-m acc	mnli-mm acc	stsb pearson	Avg.
EN-clean	90.94	40.26	87.4	84.98	84.47	65.34	77.84	77.96	82.67	76.87
syn-EN_hi-unfiltered	84.61	31.1	81.78	79.35	81.44	63.3	72.94	73.16	78.9	71.84
syn-EN_hi-unfiltered + 10%	87.39	34.22	85.77	80.96	81.07	65.11	74.76	74.38	80.32	73.78
syn-EN_hi-filtered	88.3	34.03	**86.55**	83.59	83.64	63.17	75.6	75.41	81.1	74.60
syn-EN_hi-filtered + 10%	**90.13**	**35.75**	86.41	**84.75**	**84.21**	**65.34**	**76.99**	**76.91**	**81.95**	**75.83**

Table 8: Results on English: Dev set of GLUE tasks for different synthetic splits on the *base-1k* model. Synthetic LMs perform almost as well as clean LMs after filtering and further training with clean data. **Bold** values represent the best amongst synthetic splits.

Model	sst2 acc	cola mcc	mrpc f1	qnli acc	qqp f1	rte acc	mnli-m acc	mnli-mm acc	stsb pearson	Avg.
BI-EN-HI-clean	89.56	38.53	85.56	84.88	84.39	64.25	76.4	77.27	82.07	75.88
BI-EN-HI_syn-parallel-filtered	89.56	**39.57**	85.71	84.75	**84.62**	64.98	**77.31**	**77.85**	82.41	76.31
BI-EN-HI_syn-nonparallel-filtered	**89.79**	38.68	**86.92**	**85.08**	84.06	65.34	77.15	77.55	**83.01**	**76.40**
BI-EN_syn-HI_syn-filtered	87.95	30.05	84.9	83.7	83.97	63.89	75.63	76.24	82.24	74.29
BI-EN_syn-HI_syn-filtered + 10%	89.1	35.45	85.34	84.53	84.18	**65.7**	76.64	77.24	82.1	75.59

Table 9: Results on English for Bilingual models: Dev set of GLUE tasks for different synthetic splits on the *base-1k* model. Training bilingual models using synthetic data in one language (Hindi) does not affect the performance in the other language (English). **Bold** values represent the best amongst synthetic splits.

Model	FLORES		
	EN-HI	HI-EN	Avg.
BI-EN-HI-clean	46.56	51.7	49.13
BI-EN-HI_syn-parallel-filtered	44.12	50.64	47.38
BI-EN-HI_syn-nonparallel-filtered	**45.65**	**51.29**	**48.47**
	EN-GU	GU-EN	Avg.
BI-EN-GU-clean	26.44	35.3	30.87
BI-EN-GU_syn-parallel-filtered	**26.77**	34.84	30.81
BI-EN-GU_syn-nonparallel-filtered	26.7	**36.54**	**31.62**

Table 10: chrF++ Scores on FLoRes translation task. EN-HI models are based on *base-1k* and EN-GU models are based on *mini-1k*

Perplexity-IN22 Conv				
Data	Marathi		Gujarati	
	Gemma-2B	Llama-3-8B	Gemma-2B	Llama-3-8B
Base model	332.9036	121.6586	131.8027	3.3922
clean	58.3804	15.1342	12.3128	2.5208
synthetic-unfiltered	**168.0933**	23.018	12.7995	3.0671
synthetic-filtered	191.1506	**21.355**	**11.6916**	**2.3849**

Table 11: Perplexity on IN-22 Conv. Bold values represent best among synthetic splits.

Perplexity-IN22 Gen				
Data	Marathi		Gujarati	
	Gemma-2B	Llama-3-8B	Gemma-2B	Llama-3-8B
Base model	24.893	11.821	10.4693	2.2853
clean	16.8172	7.2572	8.3868	2.1032
synthetic-unfiltered	**17.5333**	8.3766	9.0821	2.564
synthetic-filtered	17.1451	**7.8885**	**8.6089**	**2.0868**

Table 12: Perplexity on IN-22 Gen. Bold values represent best among synthetic splits.

B.2 Training

For the pretraining of the base models, we keep a hard limit for the *base-1k* model as 2.38B tokens and for the *mini-1k* model as 1B tokens. But for the TinyLM we relax this token limit until we see overfitting. For our experiments, we use the NVIDIA A100-SXM4-80GB GPUs.

B.3 Extended pretraining

For the *mini-1k* models, we randomly sample 100M tokens from the clean subset of IndicMonoDoc for the target language, and for the *base-1k* model, we sample 200M for extended pretraining. We use the same hyperparameters as training and perform extended pretraining for 2 epochs over this newly sampled clean data. For scaling experiments, we utilize TorchTune[17] for fine-tuning Llama-3-8B and Gemma-2B models. For a fair comparison, we limit each data split to 344M tokens for Gujarati and 465M tokens for Marathi and follow a similar procedure as described in Section 3.4 to generate and filter data for Marathi. We perform extended training for a single epoch using LoRA (Hu et al., 2021) finetuning on W_q, W_v projection matrices using α=16 and r=8. We keep the learning rate at $3e^{-5}$ with a weight decay of 0.01 and an effective batch size of 58k. We use the AdamW optimizer (Loshchilov and Hutter, 2019) with 1000 warmup steps and a cosine learning rate scheduler.

[17] https://github.com/pytorch/torchtune

Model	NLU						NLG				
	iXNLI	bbc-a	iitp-mr	iitp-pr	midas	Avg.	Headline Gen.	Sentence Summ.	Question Gen.	Wikibio	Avg.
HI-clean	73.61	81.75	72.58	79.73	80.34	77.60	27.54	23.64	24.84	52.16	32.04
syn-HI_en-unfiltered	72.87	77.92	64.36	76.22	79.91	74.26	27.29	22.93	24.22	50.14	31.14
syn-HI_en-unfiltered+10% clean	74.63	78.36	67.75	77.46	80.17	75.67	26.98	23.20	24.76	**51.34**	31.57
syn-HI_en-filtered	**74.75**	**81.06**	69.03	78.58	79.73	76.63	27.15	23.10	24.41	49.88	31.13
syn-HI_en-filtered+10% clean	74.49	80.94	71.61	**79.92**	**80.64**	**77.52**	**27.87**	24.23	**24.87**	51.18	**32.04**
syn-HI_en-filtered+10% synthetic	74.95	80.83	**71.93**	**79.92**	78.73	77.27	27.64	**25.13**	23.60	43.48	29.96

Table 13: Ablation Results for Hindi using additional 10% clean vs synthetic data: NLU/NLG tasks on *base-1k* (Hindi) different clean and synthetic splits. Test accuracy for NLU tasks; Rouge-L F1 scores for NLG tasks. **Bold** values represent the best amongst synthetic splits.

Model	IN22-Conv		IN22-Gen		FLORES	
	EN-HI	HI-EN	EN-HI	HI-EN	EN-HI	HI-EN
BI-EN-HI-clean	41.22	50.3	43.49	47.83	46.56	51.7
BI-EN-HI_syn-parallel-filtered	**41.92**	**49.67**	41.61	46.95	44.12	50.64
BI-EN-HI_syn-nonparallel-filtered	40.74	49.54	**42.28**	**47.66**	**45.65**	**51.29**
	EN-GU	GU-EN	EN-GU	GU-EN	EN-GU	GU-EN
BI-EN-GU-clean	35.85	41.27	22.95	31.83	26.44	35.3
BI-EN-GU_syn-parallel-filtered	34.36	41.86	22.93	30.84	**26.77**	34.84
BI-EN-GU_syn-nonparallel-filtered	**34.49**	**42.08**	**23.06**	**32.81**	26.7	**36.54**

Table 14: chrF++ Scores on FloRes, IN22-Conv and IN22-Gen splits for translation task. EN-HI models are based on *base-1k* and EN-GU models are based on *mini-1k*. **Bold** values represent the best amongst synthetic splits.

Model	IN22-Conv		IN22-Gen		FLORES	
	EN-HI	HI-EN	EN-HI	HI-EN	EN-HI	HI-EN
BI-EN-HI-clean	19.58	23.01	17.23	19.72	21.8	21.73
BI-EN-HI_syn-parallel-filtered	**19.64**	**23.79**	**16.57**	**20.14**	**21.63**	**22.6**
BI-EN-HI_syn-nonparallel-filtered	19.25	22.47	16.37	19.74	21.51	21.74
	EN-GU	GU-EN	EN-GU	GU-EN	EN-GU	GU-EN
BI-EN-GU-clean	10.24	15.19	4.65	7.92	5.44	9.57
BI-EN-GU_syn-parallel-filtered	**11.24**	**15.7**	4.87	8.44	**6.7**	10.02
BI-EN-GU_syn-nonparallel-filtered	10.86	15.57	**5.07**	**9.07**	6.17	**10.03**

Table 15: BLEU Scores on FloRes, IN22-Conv and IN22-Gen splits for translation task. EN-HI models are based on *base-1k* and EN-GU models are based on *mini-1k*. **Bold** values represent the best amongst synthetic splits.

Model	Headline Generation	Sentence Summarization	Question Generation	Wikibio Generation
BI-EN-HI-clean	27.47	23.78	24.25	50.82
BI-EN-HI_syn-parallel-filtered	26.96	**23.10**	**25.38**	48.26
BI-EN-HI_syn-nonparallel-filtered	**27.32**	22.84	24.95	**50.22**

Table 16: Performance of Bilingual models on IndicNLG tasks. All the results reported here are on base-1k and use Rouge-L F1 scores. **Bold** values represent the best amongst synthetic splits.

IndicXNLI 10-Shot				
Data	Marathi		Gujarati	
	Gemma-2B	Llama-3-8B	Gemma-2B	Llama-3-8B
Base model	**36.56 ± 0.47**	50.66 ± 1.46	**34.74 ± 0.83**	47.63 ± 1.96
clean	35.92 ± 0.79	48.47 ± 1.44	33.40 ± 0.29	48.01 ± 0.99
synthetic-unfiltered	35.66 ± 0.93	49.50 ± 1.76	33.26 ± 0.39	**48.22 ± 1.88**
synthetic-filtered	34.4 ± 0.23	**51.98 ± 1.32**	33.78 ± 0.22	47.41 ± 2.17

Table 17: Classification Results on IndicXNLI using 10-shot prompting.

Hyperparameter	Value
vocab_size	56000
val_every	0.05
bs	48
n_embed	768
num_blocks	4
num_heads	16
head_size	n_embed // num_heads
context_len	1024
block_size	context_len
attn_drop_value	0.1
dropout	0.1
ffn_drop_value	0.1
use_flashattn	TRUE
ffn_scaling	4
positional_embedding	rope'
rotatory_embedding_dim	head_size // 2
lr	6.00E-04
wd	1.00E-01
beta_1	0.9
beta_2	0.95
eps	1.00E-05
epochs	2
precision	bf16
accumulate_grad_batches	8
gradient_clip_val	1
strategy	ddp'
accelerator	gpu'
warmup_steps	5000
num_workers	16
SHUFFLE_SEED	42
PIN_MEMORY	TRUE
NUM__NODES	1
NUM_DEVICES	2

Table 18: Hyperparameters used for training the *mini-1k* model

Hyperparameter	Value
vocab_size	56000
val_every	0.05
bs	48
n_embed	768
num_blocks	12
num_heads	12
head_size	n_embed // num_heads
context_len	1024
block_size	context_len
attn_drop_value	0.1
dropout	0.1
ffn_drop_value	0.1
use_flashattn	TRUE
ffn_scaling	4
positional_embedding	rope'
rotatory_embedding_dim	head_size // 2
lr	6.00E-04
wd	1.00E-01
beta_1	0.9
beta_2	0.95
eps	1.00E-05
epochs	2
precision	bf16
accumulate_grad_batches	8
gradient_clip_val	1
strategy	ddp'
accelerator	gpu'
warmup_steps	5000
num_workers	16
SHUFFLE_SEED	42
PIN_MEMORY	TRUE
NUM__NODES	1
NUM_DEVICES	2

Table 19: Hyperparameters used for training the *base-1k* model

B.4 Fine-tuning

For GLUE tasks we use the dev split on the clean part and do hyperparameter tuning to achieve the best scores, and then we use the same hyperparameters for all synthetic experiments. For IndicGLUE we follow a similar setting for the val split to find good hyperparameters and report results on the test split like Kakwani et al. (2020a). For all classification and regression tasks, we use a single linear layer and use an appropriate activation function for classification and regression respectively. We use an Adam optimizer (Kingma and Ba, 2015) with a learning rate of $1e^{-5}$ and a batch size of 48. For NLG tasks we do extended pretraining using a separator token in between the input and output sequence with an effective batch size of 768 examples and only calculate loss for the output sequence. We use an AdamW optimizer (Loshchilov and Hutter, 2019) with learning rate = $6e^{-4}$, weight decay = $1e^{-1}$, β_1 = 0.9, β_2 = 0.95 and $\epsilon = 1e^{-5}$. For translation, we randomly sample 1M parallel sentence for each language pair from the samanantar corpus (Ramesh et al., 2022) and evaluate on FloRes (Team et al., 2022), IN22-Conv and IN22-Gen (Gala et al., 2023). We list the batch size and number of epochs of each task in Table 20.

B.5 Prompting

We use random sampling from validation sets whenever available and utilize other examples from the test set otherwise. We take 5 random samples for each evaluation of NLG tasks and 10 random samples for each evaluation of NLU tasks. We list down the prompt used below for Marathi evaluations, we use similar prompts for Gujarati as well.

```
Translate the following English sentence into मराठी.

English: The Triathlon made its Olympic debut with the
women's race.
मराठी: महिलांच्या शर्यतीद्वारे ट्रायथलॉनने ऑलिम्पिकमधील आपले पदार्पण केले.

English: Meghalaya cuisine is unique and different from
other Northeastern Indian states.
मराठी:
```

Figure 5: Prompt used for English→Marathi translation.

B.6 Evaluation

We use torch metrics[18] to calculate accuracy, f1-score, Pearson correlation, Matthew's correlation

[18] https://lightning.ai/docs/torchmetrics/stable/pages/lightning.html

Task	Batch size	Epochs	Metric
IndicXNLI	48	5	Accuracy
BBC-Articles	24	20	Accuracy
IITP-MR	24	20	Accuracy
IITP-PR	48	20	Accuracy
MIDAS	48	20	Accuracy
Headline Generation	768	2	Rouge-L F1
Sentence Summarization	768	2	Rouge-L F1
Question Generation	768	2	Rouge-L F1
WikiBio Generation	768	4	Rouge-L F1
iNLTK	48	20	Accuracy
sst2	48	10	Accuracy
CoLA	48	30	MCC
mrpc	48	30	F1
qnli	48	10	Accuracy
qqp	48	5	F1
rte	48	30	Accuracy
mnli-matched	48	5	Accuracy
mnli-mismatched	48	5	Accuracy
stsb	48	20	Pearson
XLSum Headline Gen.	768	4	Rouge-L F1
XLSum Question Gen.	768	4	Rouge-L F1
CNN Dailymail	768	4	Rouge-L F1
DialogSum	768	4	Rouge-L F1
Samanantar	768	2	chrF++ / BLEU

Table 20: Hyperparameters used for finetuning tasks

coefficient. We report chrF++ scores[19] and BLEU scores[20] (Papineni et al., 2002) using the sacreBLEU[21] implementation and Rouge-L f1 scores using the sacreRouge (Deutsch and Roth, 2020) implementation by the xl-sum repository[22].

We report English scores for NLU on the validation split of the GLUE benchmark and test splits for XL-Sum, CNN Dailymail, and Dialogsum NLG benchmarks. For Hindi and Gujarati, we use the test split of IndicGLUE and IndicXNLI.

For classification and regression tasks, we use the models finetuned according to hyperparameters mentioned in Appendix B.4 to keep fair comparison for all models and mention results on the final epoch. For generations on IndicNLG and English NLG tasks, we use beam search with a beam width of 5, length penalty of 1.0, n_gram repetition penalty of 4 n_grams with sampling set to false and early stopping set to true. We also set a maximum generation length to 64 tokens. For the translation task, we follow a beam search with a beam width of 5, maximum new tokens to 256 and early stopping to true.

C Utilising Translationese

In this section we provide details on the generation and filtering of translationese data for our experiments.

C.1 Creating synthetic data

"Translationese" is a term used to describe peculiarities in the text translated into a specific language, differentiating it from content originally written in that language (Gellerstam, 1986). Translated texts into the target language (via humans or machine-generated) often show distinctive features that differentiate them from their original counterparts in the target language. These disparities arise from either the influence of the translation process itself on the final product or the inherent "fingerprints" of the source language subtly present in the target language rendition (Rabinovich and Wintner, 2015). This is a common phenomenon in translation models where the target language translations often show characteristics of the source language and add bias to the evaluation of downstream tasks (Toral et al., 2018; Zhang and Toral, 2019; Graham et al., 2019). So far a lot of work on synthetic translated data has been done for using back translations (Sennrich et al., 2016a; Edunov et al., 2018) for improving Machine translation performance (Marie et al., 2020; Bogoychev and Sennrich, 2019; Ni et al., 2022) or for classification tasks like native language identification (Goldin et al., 2018), etc. Tranlationese data has been used for many tasks but we explore the efficacy of using translationese data for pretraining of language models. We collect monolingual corpora in the source language as mentioned in Section 3.1 and utilize a powerful off-the-shelf translation model IndicTrans2 (Gala et al., 2023) to generate translationese data. Since IndicTrans2 can only handle a max sentence length of 256 BPE tokens, we split the documents using Moses Sentence Splitter[23] to perform translations into the target language at the sentence level and then merge again to form documents. We also repair translations that exceed in length 256 BPE tokens using the TinyLM trained on clean corpora as mentioned in Section 4 to complete the sentence translation, we encounter only 0.002% of such cases. We use this corpus for the *synthetic* and *clean+synthetic* part of our experiments.

C.2 Perplexity filtering

Following Figure 2, we use these TinyLMs to filter the generated synthetic translationese corpora from IndicTrans2. We do this by using perplexity as a

[19] chrF++ signature nrefs:1|case:mixed|eff:yes|nc:6|nw:2|space:no|version:2.4.0

[20] sacreBLEU signature: nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.4.0

[21] https://github.com/mjpost/sacrebleu

[22] https://github.com/csebuetnlp/xl-sum

[23] https://pypi.org/project/mosestokenizer/

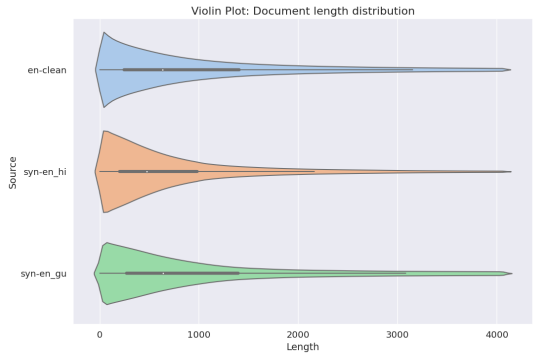

Figure 6: Violin plot displaying the distribution of lengths of clean and filtered English documents on different data splits: *en-clean* (English web documents), *syn-en_hi* (synthetic English documents translated from Hindi), and *syn-en_gu* (synthetic English documents translated from Gujarati).

measure of document quality score. For language models, perplexity quantifies how well a model predicts a sequence of tokens. A lower perplexity indicates better predictive performance. While calculating perplexity over a sequence of tokens $W \in w_1, w_2, \ldots, w_N$ we skip the first s tokens where $s = 10$, $e = 1024$ and calculate loss until only the first e tokens of the document. We find setting e to larger values can lead to higher variance in the document scores due to the size of the TinyLM. After initial analysis, we choose s and e such that we remove the high uncertainty of the language at the start of an unseen document and avoid penalizing longer documents due to the fragility of the extrapolation ability of TinyLM[24]. Note that it is important to choose e such that the language model gives a uniform estimate of perplexity over an already seen sequence of tokens $\in w_s, w_{s+1}, \ldots, w_e$. For our experiments, we use the TinyLMs to score all synthetically generated translationese data and calculate a document score using the above method. Following Laurençon et al. (2022), we do subsampling by thresholding document perplexity scores except Laurençon et al. (2022) did it using Ken-LM (Heafield, 2011) and we do it using our TinyLM. We keep the threshold value such that we include enough documents to reach the computed optimal token count for pretraining experiments.

[24]During experiments we saw that these TinyLMs can only go up to a certain context length before deteriorating in quality.

High Resource		Low Resource	
Lang	**#Tokens**	**Lang**	**#Tokens**
bn	5,258.47	as	57.64
en	11,986.53	brx	2.25
gu	887.18	doi	0.37
hi	11,268.33	gom	2.91
kn	567.16	kas	1.27
ml	845.32	mai	1.51
mr	1,066.76	mni	0.99
ne	1,542.39	or	81.96
pa	449.61	sa	80.09
ta	2,171.92	sat	3.05
te	767.18	sd	83.81
ur	2,391.79		

Table 21: Languagewise corpora size in Million tokens

D Qualitative Analysis

Since translation errors occur frequently, leading to biased, ungrammatical, or erroneous translations that can have drastic consequences on training, it is important to mitigate or remove such errors in translationese corpora. The most common machine translation errors include mistranslations due to ambiguous words, incorrect handling of expressions, syntax and grammar errors, and issues with preserving context across longer sentences. Many approaches have been proposed to address these errors, but most are computationally expensive, especially when translationese data is used for pretraining. Instead, we ask whether a language model can identify such issues. To investigate this, we examine which types of English sentences were filtered.

In many cases, we found that the filtered documents included errors like code-mixing and repetitions, often generated by the diverging output of the translation model. This is expected since such phenomena are rarely seen in natural written language, and the model assigning high entropy suggests an unlikely sequence outcome. Although the model regarded some erroneous instances as false positives, many such cases were successfully avoided. We also noticed that, due to the small size of the model, many perfectly good documents were filtered out because of the model's inability to evaluate them. This issue was observed less frequently in larger models used for evaluation. As seen in Figure 7, the filtering model fails to understand complex words and named entities, which it regarded as unlikely due to its preference for sim-

pler terms and more likely entities. Other common elements that were filtered included numbers, dates, and abbreviations. While this can lead to the loss of valuable information, as many good documents are discarded, we empirically observe the benefits of such filtering. These errors could likely be reduced by using a larger filtering model that can better approximate the source language but we leave this analysis for future work.

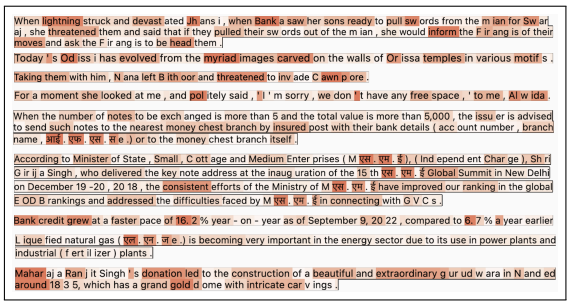

Figure 7: Heatmap of perplexity over filtered sentences.

E IndicMonoDoc

In this section, we describe the process of creating the IndicMonoDoc corpus which is the largest document-level corpora for Indic languages consisting of 39.5 billion tokens spanning 23 languages. IndicMonoDoc comprises 27.5B Indic tokens and 12B tokens of English tokens. Table 21 shows language language-wise deduplicated size of the IndicMonoDoc corpus and Figure 3 shows a comparative 100% stacked bar plot with IndicCorpv2 which is a sentence level corpora.

E.1 Crawling

To extract URLs from the web we sample word level *n-grams*; *n=2,...,6* from a sample monolingual corpora to create a list of keyword searches. We then randomly merge *k*; *k=1,..,4* keywords to form a query. Using these queries we perform automatic web searches to collect a large repository of URLs. We merge this list with a manual list of sources to perform URL-level deduplication. We crawl these webpages leaving out some of them[25]. We leave out webpages that consist of a considerable amount of English content using a simple script recognition regex. We perform this scrapping majorly for the bottom 14 low-resource languages. We also add script-level recognition using Unicode characters[26] for each language before crawling a webpage to avoid scrapping non-Indic text.

E.2 Post processing

A lot of crawled content consists of unwanted text like HTML tags, emoticons, and text in another language. We use manual filtering pipelines inspired by OSCAR (Ortiz Suárez et al., 2019), (Abadji et al., 2022) to remove such content. We additionally use a language detection-based (LID) filtering using cld3[27] and IndicLID-FTN model (Madhani et al., 2023a) to discard languages not of interest. Following Doddapaneni et al. (2023) we perform document filtering to remove offensive text from the corpora using a list of offensive words and phrases extended from work by Team et al. (2022) which consists of offensive words in 209 languages. We also use a Romanized version of this list using the transliteration tool by Madhani et al. (2023b) to perform toxic document filtering in 17 languages. Following Kakwani et al. (2020a) & Doddapaneni et al. (2023) we merge all the filtered corpus with Wikipedia, OSCAR (Ortiz Suárez et al., 2019) and some dumps of mC4 (Xue et al., 2021). Finally, we perform deduplication at paragraph level using Murmurhash algorithm[28] with a 128-bit unsigned hash for each monolingual split of the corpora. After all post-processing steps, the language wise size of the corpora is mentioned in Table 21. A major chunk of the corpus is comprised of English, Hindi, and Bengali which make up 72.15% of the corpora.

[25] We leave webpages consisting of a robots.txt file and URLs containing offensive text or social media links

[26] https://unicode.org/charts/

[27] https://github.com/google/cld3

[28] https://pypi.org/project/mmh3/