IndicGenBench: A Multilingual Benchmark to Evaluate Generation Capabilities of LLMs on Indic Languages

Harman Singh, Nitish Gupta, Shikhar Bharadwaj, Dinesh Tewari, Partha Talukdar
Google Research India
{hrman, guptanitish, shikharop, dineshtewari, partha}@google.com

Abstract

As large language models (LLMs) see increasing adoption across the globe, it is imperative for LLMs to be representative of the linguistic diversity of the world. India is a linguistically diverse country of 1.4 Billion people. To facilitate research on multilingual LLM evaluation, we release IndicGenBench — the largest benchmark for evaluating LLMs on user-facing generation tasks across a diverse set 29 of Indic languages covering 13 scripts and 4 language families. IndicGenBench is composed of diverse generation tasks like cross-lingual summarization, machine translation, and cross-lingual question answering. IndicGenBench extends existing benchmarks to many Indic languages through human curation providing multi-way parallel evaluation data for many under-represented Indic languages for the first time. We evaluate a wide range of proprietary and open-source LLMs including GPT-3.5, GPT-4, PaLM-2, mT5, Gemma, BLOOM and LLaMA on IndicGenBench in a variety of settings. The largest PaLM-2 models performs the best on most tasks, however, there is a significant performance gap in all languages compared to English showing that further research is needed for the development of more inclusive multilingual language models. IndicGenBench is available at www.github.com/google-research-datasets/indic-gen-bench

1 Introduction

With the advances in generative language technologies powered by Large Language Models (LLMs; Brown et al., 2020; Rae et al., 2021; Chowdhery et al., 2022; OpenAI et al., 2023; Tay et al., 2023; Google, 2023), there has been a surge of interest in evaluating the multilingual capabilities of these models. Recent work (Ahuja et al., 2023a,b) shows a consistent performance gap between high resource languages and languages with lower amounts of web resources available. To develop highly multilingual generative LLMs which should work equally well for 100s of languages spoken by billions of people in the world, it is crucial to evaluate their capabilities across a variety of languages to uncover performance gaps and guide future research.

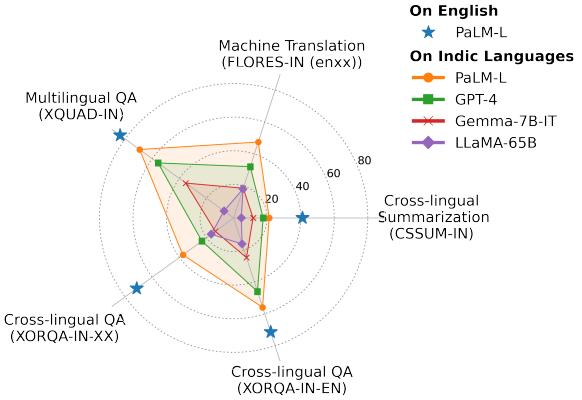

Figure 1: **Performance of state-of-the-art LLMs on different tasks in IndicGenBench**. We observe a significant performance gap between English and Indic languages across LLMs.

In this work we focus on India, a country with 1369 rationalized mother tongues spoken by more than a billion people.[1] Making progress on language technologies for Indic languages will not only improve the state of affairs in this region, but will also provide valuable learning to the NLP community which will be applicable to other geographical regions and language families. There are has been much work from the community in building natural language *understanding* (NLU) models for Indic languages (Kakwani et al., 2020; Khanuja et al., 2021), as well as evaluation datasets (Dodda-

All authors are now part of Google DeepMind

[1] https://en.wikipedia.org/wiki/Languages_of_India

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 11047–11073

Task	Language	Input	Output	#Languages (H / M / L)	Dataset Size (Train / Dev / Test)
CrossSum-In (Cross-lingual Summarization)	Hindi	**Article:** Nasa said a problem with a seismometer provided by the French space agency meant the launch could not go ahead. It is feared the mission could now face a two-year delay. The InSight spacecraft was scheduled to take off between 4-30 March and land on the Red Planet six months later to examine Mars' geology in depth. Nasa said it had decided to call off the launch because the agency was unable to fix a leak affecting the seismometer, which required a vacuum seal to cope with harsh [.....]	अमरीकी अंतरिक्ष एजेंसी नासा अपने अंतरिक्ष यान डिस्कवरी में आई उस समस्या की जाँच कर रही है जिसकी वजह से बुधवार को निर्धारित उड़ान स्थगित करनी पड़ी.	9 / 7 / 13	2.9k / 2.9k / 14.5k
Flores-In (Machine Translation)	Konkani	She gained critical acclaim during her time in Atlanta and was recognized for innovative urban education.	अटलांटात तिच्या कारकिर्दीत तिका समिक्षकांची तोखणाय मेळ्ळी आनी नाविन्यपुर्ण शारी शिक्षणा खातीर तिका नामनां मेळ्ळीं	9 / 7 / 13	- / 28.9k / 29.3k
XQuAD-In (Multilingual QA)	Punjabi	**Question:** ਓਪਰੇਸ਼ਨ ਗਲੋਰੀ ਦਾ ਮਕਸਦ ਕੀ ਸੀ? **Context:** ਜੰਗ ਤੋਂ ਬਾਅਦ, ਜੁਲਾਈ ਤੋਂ ਨਵੰਬਰ 1954 ਤੱਕ ਓਪਰੇਸ਼ਨ ਗਲੋਰੀ ਚਲਾਇਆ ਗਿਆ, ਤਾਂ ਜੋ ਜੰਗ ਵਿੱਚ ਸ਼ਾਮਲ ਦੇਸ਼ ਆਪਣੇ ਸ਼ਹੀਦਾਂ ਦੀਆਂ ਲਾਸ਼ਾਂ ਨੂੰ ਆਪਣੇ ਦੇਸ਼ ਵਾਪਸ ਲਿਆ ਸਕਣ। ਇਸ ਤਹਿਤ ਯੂ.ਐੱਸ. ਫ਼ੌਜ ਅਤੇ ਯੂ.ਐੱਸ. ਸਮੁੰਦਰੀ ਫ਼ੌਜੀ ਪਲਟਨ ਦੇ 4,167 ਸਿਪਾਹੀ ਅਤੇ 13,528 KPA ਅਤੇ PVA ਦੇ ਸ਼ਹੀਦਾਂ ਦੀਆਂ ਲਾਸ਼ਾਂ ਨੂੰ ਵਾਪਸ ਲਿਆਂਦਾ ਗਿਆ। ਨਾਲ ਹੀ, ਜੰਗ ਦੌਰਾਨ ਸੰਯੁਕਤ ਰਾਸ਼ਟਰ ਦੇ ਕੈਦੀਆਂ ਦੇ ਕੈਂਪ ਵਿੱਚ ਮਰਨ ਵਾਲੇ 546 ਨਾਗਰਿਕਾਂ ਨੂੰ ਦੱਖਣੀ ਕੋਰੀਆ ਦੀ ਸਰਕਾਰ ਨੂੰ ਸੌਂਪਿਆ ਗਿਆ ਸੀ। [........]	ਜੰਗ ਵਿੱਚ ਸ਼ਾਮਲ ਦੇਸ਼ ਆਪਣੇ ਸ਼ਹੀਦਾਂ ਦੀਆਂ ਲਾਸ਼ਾਂ ਨੂੰ ਆਪਣੇ ਦੇਸ਼ ਵਾਪਸ ਲਿਆ ਸਕਣ	9 / 3 / -	1.2k / 1.2k / 14.3k
XorQA-In-Xx (Cross-lingual QA)	Telugu	**Question:** 1989 లో భారత ప్రధాని ఎవరు? **Context:** Rajiv Ratna Gandhi (; 20 August 1944 – 21 May 1991) was an Indian politician who served as the 6th Prime Minister of India from 1984 to 1989. He took office after the 1984 assassination of his mother, Prime Minister Indira Gandhi, to become the youngest Indian Prime Minister at the age of 40.	రాజీవ్ రత్న గాంధీ	9 / 6 / 13	2.8k / 14k / 15.1k
XorQA-In-En (Cross-lingual QA)	Santali	**Question:** [illegible] **Context:** Tibet had been under the administrative rule of the Qing dynasty since 1720. [.......]. In early 1912, the Government of the Republic of China replaced the Qing dynasty as the government of China and the new republic asserted its sovereignty over all the territories of the previous dynasty, which included 22 Chinese provinces, Tibet and Mongolia.	Government of the Republic of China	9 / 6 / 13	2.8k / 14k / 15.1k

Table 1: IndicGenBench, our proposed benchmark, consists of five tasks: Cross-lingual Summarization (CrossSum-In), Machine Translation (Flores-In), Multilingual QA (XQuAD-In) and Cross-lingual QA (XorQA-In-Xx and XorQA-In-En). An example from each task, the number of languages for which we collect evaluation data (divided by resourcefulness, higher (H), medium (M) and low (L)), and the number of training/validation/test instances per task is shown above. See Section 2 for details.

paneni et al., 2023; Mhaske et al., 2023) to support such models. In this work, our focus is to develop a high-quality benchmark for evaluating *generative* language capabilities in a variety of Indic languages across various levels of resourcefulness.

We release IndicGenBench, a multilingual, multi-way parallel benchmark for measuring language generation capabilities across diverse user-facing tasks in 29 Indic languages across 4 language families (Table 7). IndicGenBench extends existing benchmarks such as CrossSum (Bhattacharjee et al., 2023), XQuAD (Artetxe et al., 2020), XorQA (Asai et al., 2021), and FLORES (NLLB-Team et al., 2022) for additional Indic languages and is composed of tasks like cross-lingual summarization (CrossSum-In), machine translation (Flores-In), cross-lingual reading comprehension (XorQA-In-Xx and XorQA-In-En) and multilingual reading comprehension (XQuAD-In). Each dataset consists of parallel examples in up to 29 low to comparatively higher resource Indic languages; and for some tasks (e.g. CrossSum-In), IndicGenBench provides the first-ever evaluation set for as many as 18 of these languages. We also release a small training set in all tasks for efficient adaptation of LLMs. Our comprehensive evaluation of various state-of-the-art proprietary and open-source LLMs on IndicGenBench shows that there is a significant gap in performance between English and Indic languages (see Figure 1). Our contributions are as follows:

- Created and released IndicGenBench, a high quality text benchmark in diverse language generation tasks like summarization, question-answering, and translation across 29 Indic languages. IndicGenBench is the largest generation benchmark for Indic languages.
- Comprehensive experimentation on SoTA LLMs (mT5, Gemma, BLOOM, LLaMA, GPT-3.5, GPT-4, PaLM-2) across various model sizes and training settings to benchmark their Indic language generation capabilities.
- A qualitative analysis for assessing the gaps in current language technologies and define potential directions of future research.

2 INDICGENBENCH

INDICGENBENCH is a high-quality, human-curated benchmark to evaluate text generation capabilities of multilingual models on Indic languages. Our benchmark consists of 5 user-facing tasks (viz., summarization, machine translation, and question answering) across 29 Indic languages spanning 13 writing scripts and 4 language families. For certain tasks, INDICGENBENCH provides the first-ever evaluation dataset for up to 18 Indic languages. Table 1 provides summary of INDICGENBENCH and examples of instances across tasks present in it.

Languages in INDICGENBENCH are divided into (relatively) Higher, Medium, and Low resource categories based on the availability of web text resources (see appendix §A for details).[2]

Higher (9): Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Tamil, Telugu, Urdu
Medium (7): Assamese, Bhojpuri, Nepali, Odia, Punjabi, Pashto, Sanskrit
Low (13): Awadhi, Haryanvi, Tibetan, Garhwali, Konkani, Chhattisgarhi, Rajasthani, Maithili, Manipuri, Malvi, Marwari, Santali, Bodo

As evident from the lists above, our benchmark provides a broad-coverage over languages with respect to their resourcedness, allowing users to evaluate language models on relatively high-resource languages such as Hindi and extremely low-resource languages such as Manipuri in Meitei script on a single benchmark.

To curate the evaluation datasets for our benchmark, we use the following existing datasets as the source: CrossSum (Bhattacharjee et al., 2023) for cross-lingual summarization, FLORES (NLLB-Team et al., 2022) for machine translation, XQuAD (Artetxe et al., 2020) for multilingual QA, and XoRQA (Asai et al., 2021) for cross-lingual QA. From each of these datasets we select a subset of English examples to be a part of our benchmark, and then collect professional human translations for these examples in all target Indic languages. Some target languages are already covered by the source datasets in which case we re-purpose this existing data and only collect translations for the remaining languages. We also collect and release a small amount of training and validation data making possible evaluation of training techniques like fine-tuning, parameter-efficient training, in-context learning, and others.

Why extend existing benchmarks? We chose to collect human translations of existing benchmarks as opposed to creating evaluation data from scratch due to various reasons:

- Translation-based extension of existing benchmark results in multi-way parallel data, allowing researchers to attribute performance due to task knowledge vs. language understanding, and measure cross-lingual generalization
- For many low-resource languages in INDICGENBENCH, clean text knowledge corpus (e.g., Wikipedia) is not available making it difficult to acquire source data for annotation
- By focusing only on translation quality in the target Indic languages, we are able to leverage the quality control that went into designing the source benchmarks.

Annotators were professional data labelers working as contractors at our organization and with a vendor. Annotators were paid competitive rates in compliance with applicable labor laws and prevailing market rates. Our pay rate to annotators varied across languages, ranging from USD 2.80 per hour for Pashto to USD 15.90 per hour for Tibetan.

Cross-Lingual Summarization: CROSSSUM-IN

We create CROSSSUM-IN based on CrossSum (Bhattacharjee et al., 2023), a dataset for cross-lingual summarization, which in turn is derived from XL-Sum (Hasan et al., 2021b). CrossSum contains multi-way parallel data in 45 languages where BBC news articles as source in a language are paired with corresponding summaries in other languages. Based on their matching criteria, different languages have different amount of source-target pairs.

We sample 700 English article-summary pairs (100 each from train/dev and 500 from test) and ask human translators to translate the English summary into the target Indic languages. CrossSum already contains data for 9 of our 29 target languages; for these languages we sample 100/100/500 examples from the original dataset to maintain equity with other languages we collect data for. CROSSSUM-IN contains a total of 20.3k examples across 29 Indic languages in our benchmark.

[2]We note that the languages called relatively higher resource in this paper, e.g., Hindi or Bengali, are in fact mid-low Web resource when compared to English and other truly high resource languages. For example, using Wikipedia as a proxy for language resources, compared to 6.6M+ Wikipedia articles in English, there are only 160K Hindi Wikipedia articles.

Machine Translation: FLORES-IN

FLORES-200 (NLLB-Team et al., 2022) is a human-annotated multi-way parallel machine translation (MT) benchmark for 200 languages where the same source English sentences are translated by humans into the target 200 languages. It contains data in 22 of our 29 target languages; we extend this by collecting human translations for the remaining 7 new languages leading to a MT benchmark in 29 Indic languages which we call FLORES-IN.

FLORES-200 is divided into three splits: dev (997), devtest (1012), test (992), of which the test set it not public. We collect translations for all 997 dev and 1012 devtest sentences, yielding 2009 sentences per language. Collectively, FLORES-IN contains 58.2k examples across 29 Indic languages.

Multilingual Question-Answering: XQUAD-IN

We create an Indic Multilingual Question Answering task XQUAD-IN based on the multilingual reading comprehension dataset XQuAD (Artetxe et al., 2020). XQuAD is in turn derived from the SQuAD dataset (Rajpurkar et al., 2016), in which an English Wikipedia passage is paired with multiple question-answer (QA) pairs where the answers are short spans for the given passage. The authors of XQuAD collected human translations for 240 passages and 1190 QA pairs from the SQuAD v1.1 development set into 10 higher resource languages (Hindi being the only Indic language).

To create XQUAD-IN, we use the 240 passages and 1190 QA pairs from XQuAD as our test set. We additionally selected 20 passages and 100 QA pairs from the original SQuAD v1.1 training and development sets each to create our training and development set. For all the 280 passages and 1390 QA pairs we collect professional human translations in 12 Indic languages.[3] Overall, XQUAD-IN contains 3.3k passages and 16.6k QA pairs in 12 Indic languages.

Cross-Lingual Question-Answering: XORQA-IN

We create Indic Cross-lingual Question-Answering dataset XORQA-IN based on the XOR-TYDI QA dataset (Asai et al., 2021). XOR-TYDI contains questions in non-English languages paired with English evidence passages and short span answers from those passages (similar to SQuAD). It was created with the idea of developing NLP systems that can answer questions in users' native language by refering to sources in a high-resource language, such as English, which was more likely to contain the answer due to the *information scarcity* of low-resources languages on the web. The original XOR-TYDI contains data in 7 languages out of which Bengali and Telugu are the two Indic languages.

To create XORQA-IN, we select the 302 Bengali and 237 Telugu examples (Bn/Te-question, En-passage, En-answer) from the XOR-TYDI dev set as our test data.[4] Additionally, we sample 600 examples (equally from Bengali and Telugu) from the training set of XOR-TYDI to create our training (100) and development (500) set. We then follow a two-staged translation process, where we first ask the human translators to translate the Bengali or Telugu question (Bn/Te-question) into English (En-question). In the second stage, we collect translations for these English questions (En-question) into target languages (Xx-question) and translations for the English answers (En-answer) into the target languages (Xx-answer).

We create two tasks from this translated data:
1. **XORQA-IN-EN**: Each example contains (Xx-question, En-passage, En-answer). This task is similar to the XOR-TYDI dataset.
2. **XORQA-IN-XX**: Each example contains (Xx-question, En-passage, Xx-answer), where the task is to generate the answer in the same language as the question.

We collect data for 28 Indic languages resulting in 32k examples.[5]

See Appendix Table 9 for languages covered by each dataset in INDICGENBENCH.

3 Experiments and Analysis

We use INDICGENBENCH to benchmark multilingual and cross-lingual language generation capabilities of various LLMs on Indic languages. We perform experiments with a variety of open-source LLMs — mT5 (Xue et al., 2021), LLaMA (Touvron et al., 2023),[6], BLOOMZ (Workshop et al., 2022), Gemma (Team et al., 2024); and proprietary LLMs — GPT-3.5, GPT-4 (OpenAI et al., 2023), and PaLM-2 (Anil et al., 2023).

We compare and analyze the performance of different model size variants of these LLMs under var-

[3] XQUAD-IN contains all 9 higher-resource languages (see §2) and 3 medium-resources languages, namely, Assamese, Odia, and Punjabi.

[4] XOR-TYDI has not publicly released its test set.

[5] We do not collect translations for Nepali.

[6] LLaMA-2 could not be used due to a restrictive licence

Model (LLM)	CrossSum-In	Flores-In	XQuAD-In	XorQA-In-Xx	XorQA-In-En
Eval. Metric	ChrF	ChrF (enxx / xxen)	Token-F1	Token-F1	Token-F1
*Performance in **English***					
GPT-4	30.3	– / –	64.8	–	37.9
PaLM-2-L	41.1	– / –	83.7	–	71.4
Average Performance on **IndicGenBench**					
LLaMA-7B	3.7	11.5 / 21.6	3.8	7.4	10.4
LLaMA-13B	4.1	13.3 / 24.1	4.5	10.4	12.1
LLaMA-65B	4.6	18.1 / 32.7	7.1	16.5	16.3
BLOOM-7B	3.8	18.3 / 31.2	13.8	7.9	23.6
BLOOMZ-7B	1.2	40.8 / 48.4	53.7	7.0	49.0
Gemma-7B-PT	0.0	32.1 / 50.4	0.5	11.7	23.8
Gemma-7B-IT	11.6	18.6 / 29.2	35.3	13.5	24.8
GPT-3.5	16.3	29.2 / 47.7	33.2	21.6	35.5
GPT-4	17.6	32.1 / 54.5	55.7	23.4	46.0
PaLM-2-XXS	7.2	24.0 / 43.4	34.6	13.5	36.8
PaLM-2-XS	15.5	40.7 / 58.3	62.2	29.5	47.8
PaLM-2-S	18.5	43.5 / 61.6	66.7	31.6	**57.4**
PaLM-2-L	**21.2**	**47.5 / 65.1**	**69.3**	**37.4**	55.9

Table 2: **One-shot performance on IndicGenBench** across model sizes for all LLMs considered in our work (§3.1). For each LLM family performance improves with increasing model size, with PaLM-2-L performing the best across most tasks. Compared to English, all models under-perform significantly highlighting shortcomings of current SoTA LLMs. See Section 3.1 for details.

ious learning paradigm settings. We first evaluate model performance on one-shot prompting (§3.1) and also measure performance across language categories based on resourcedness (§3.2). We then evaluate the effect of number of in-context examples shown to the model as supervised data (§3.3) and the effect of prompting in a higher-resource language such an English or Hindi (§3.4). Using the training data contained in IndicGenBench, we measure how the performance of LLMs after fine-tuning compares with few-shot prompting (§3.5). Finally, we perform qualitative analysis of models on IndicGenBench and highlight some areas of improvement for future model development (§3.7).

Evaluation Metrics For the cross-lingual summarization and translation tasks, CrossSum-In and Flores-In, we report Character-F1 (ChrF) metric (Popović, 2015) since token-level metrics like ROUGE and BLEU are not reliable for low-resource languages (Bapna et al., 2022). To stay consistent with existing literature on QA tasks, we report SQuAD-style Token-F1 on our XQuAD-In and XorQA-In QA tasks.

On Flores-In, we report translation performance in both directions—translating from English to the target language (enxx) and vice-versa (xxen).

3.1 Comparison of LLMs on IndicGenBench

In Table 2 we evaluate LLaMA, BLOOMZ, Gemma, GPT and PaLM-2 family of models on all tasks of IndicGenBench in a one-shot prompted setting. Numbers are averaged across all languages in the evaluation data. To compare, we also report English performance for GPT-4 and PaLM-2-L.

We see across tasks that larger models from the same LLM family perform better. PaLM-2-L performs the best among all LLMs considered, except for the XorQA-In-En task where PaLM-2-S performs slightly better. We find that open source LLaMA models perform much worse compared to proprietary models; even the largest LLaMA-65B model significantly underperforms the smallest PaLM-2-XXS model. Gemma-7B instruction tuned model performs better than LLaMA-13B as well as LLaMA-65B on most tasks. BLOOMZ, which is an instruction tuned version of BLOOM (Workshop et al., 2022), pre-trained on large-scale multilingual data, works the best on three out of five tasks in IndicGenBench. On CrossSum-In and XorQA-In-Xx it falls behind LLaMA and Gemma. Compared to English, we see significant room for improvement (20+ ChrF or Token-F1 points) across all tasks.

3.2 Performance across language categories

In Table 3 we report one-shot performance across language categories defined in Section 2. We only show performance for Gemma-7B-IT, BLOOMZ-7B, LLaMA-65B, GPT-4 and PaLM-2-L models here and report performance for the other models in appendix B.1. We find that there is a significant performance drop going from higher resourced languages to medium resourced ones, and further drop in lower resourced languages.

We would like to point out two observations here: (a) In Flores-In, the performance for translating English to the target language (enxx) drops significantly from higher to lower resourced languages (56.9 → 41.9 for PaLM-2-L) whereas the performance in the xxen direction does not fall this drastically (68.2 → 62.6). This is also seen in XorQA-In-Xx and XorQA-In-En. This highlights that current LLMs are better at understanding than generation in these lower-resourced languages.

	CrossSum-In			Flores-In (enxx / xxen)			XQuAD-In		XorQA-In-Xx			XorQA-In-En		
Model	**High**	**Medium**	**Low**	**High**	**Medium**	**Low**	**High**	**Medium**	**High**	**Medium**	**Low**	**High**	**Medium**	**Low**
LLaMA-65B	4.4	4.6	4.7	18.2 / 31.5	15.4 / 30.0	19.5 / 35.0	8.8	1.9	17.7	13.5	17.1	16.4	14.0	17.3
Gemma-7B-IT	13.9	11.5	10.0	17.6 / 33.7	15.0 / 26.1	21.3 / 27.7	38.8	24.8	18.9	8.3	12.2	29.5	23.9	21.9
BLOOMZ-7B	1.5	1.7	0.6	**67.7** / 59.1	39.4 / 50.2	22.9 / 40.0	55.5	48.1	10.8	2.8	6.2	**64.7**	45.8	39.5
GPT-4	19.4	17.9	16.3	36.2 / 59.6	30.7 / 55.2	29.9 / 50.5	56.1	54.6	25.8	21.6	22.6	49.4	50.0	41.8
PaLM-2-L	**25.2**	**23.1**	**17.5**	56.9 / **68.2**	**45.9 / 65.6**	**41.9 / 62.6**	**72.5**	**59.8**	**41.9**	**36.7**	**34.6**	57.3	**57.9**	**53.9**

Table 3: **One-shot performance across language categories** based on resourcedness defined in Section 2. For all tasks, we witness significantly lower performances in medium and low resource languages compared to the higher resource ones. Please see Table 10 in appendix B.1 for results on other models. See Section 3.2 for more details.

(b) In few cases, we see smaller performance deltas between medium and lower resourced languages compared to higher and medium categories. From our analysis, this can mainly be attributed to many languages in the lower category being similar to Hindi and written in the same Devanagari script.

	Flores-In			XorQA-In-Xx			
Model (LLM)	**0**	**1**	**5**	**0**	**1**	**2**	**3**
LLaMA-7B	8.0	11.5	11.4	5.0	7.4	9.0	9.2
LLaMA-13B	8.6	13.3	13.4	6.3	10.4	12.2	13.1
LLaMA-65B	14.0	18.1	18.3	12.3	16.5	18.7	19.4
PaLM-2-XXS	0.8	24.0	26.9	8.9	13.5	15.8	17.5
PaLM-2-XS	20.1	40.7	42.3	21.4	29.5	32.2	33.2
PaLM-2-S	24.9	43.5	45.2	22.7	31.6	33.4	35.4
PaLM-2-L	**31.1**	**47.5**	**49.3**	**31.9**	**37.4**	**39.7**	**41.1**

Table 4: **Performance by varying number of in-context exemplars** for LLaMA and PaLM-2 models on Flores-In (enxx) and XorQA-In-Xx tasks (§3.3). Performance improves with increasing amounts of supervision provided in-context. Refer appendix B.2 for results on other tasks and models.

3.3 In-context learning on IndicGenBench

In this section we aim to understand the impact of the number of *in-context examples* shown to the LLM during few-shot prompting.

Since CrossSum-In and XQuAD-In input passages are long, we are only able to perform 0-and-1-shot prompting. For XorQA-In-Xx and XorQA-In-En we perform 0-to-3-shot prompting, and for Flores-In we perform 0, 1 and 5-shot prompting.

We show performance for Flores-In and XorQA-In-Xx in Table 4. Other results are shown in appendix B.5 due to space limitations. Across model families and sizes we observe that increasing the amount of supervision in terms of the in-context examples improves performance.

3.4 Transfer from high-resource languages

For languages with no supervised data, one option to improve performance is utilizing existing supervised data another language as in-context exemplars. In this section we aim to study if the language in which the model is prompted plays a role in performance.

In Table 5 we show performance when the model is prompted in English vs. Hindi, a representative higher resourced Indic language. For comparison, we also show performance when the in-context exemplar is in the same language as the test instance. We find that Hindi in-context exemplars are much more useful for all models as compared to their English counterparts. Surprisingly, for smaller models, performance with Hindi exemplars comes extremely close to prompting in the test language, even better sometimes.

3.5 Fine-tuning LLMs on IndicGenBench and Comparison with In-Context Learning

As outlined in Section 2, we also release a small, high-quality training set for all tasks in IndicGenBench (except Flores-In which only has dev and test sets). This training data can be used to adapt LLMs to downstream tasks in Indic languages via fine-tuning and other training techniques.

Table 6 shows our results of fine-tuning mT5 and PaLM-2 models and their comparison with in-context learning using PaLM-2. We fine-tune each model on training data from all available languages including English, use the development set for early stopping, and report numbers on the test set. For question-answering tasks that require generating short spans as answers, we find that older generation mT5 models significantly outperform smaller PaLM-2 models in most cases.[7] On CrossSum-

[7]Since the parameter count for PaLM-2 models is not

	CrossSum-In			XQuAD-In		XorQA-In-Xx			XorQA-In-En		
Model (1-Shot Lang)	**Higher**	**Medium**	**Low**	**Higher**	**Medium**	**Higher**	**Medium**	**Low**	**Higher**	**Medium**	**Low**
PaLM-2-XXS (En)	0.3	0.1	0.3	38.5	31.9	14.0	5.4	7.3	40.3	35.0	30.8
PaLM-2-XXS (Hi)	1.3	2.1	3.7	**39.8**	**33.3**	17.6	8.5	10.5	**45.5**	**39.4**	**31.9**
PaLM-2-XXS (Lang)	**7.7**	**7.6**	**6.7**	37.2	26.8	**17.7**	**8.8**	**12.8**	43.6	38.3	31.5
PaLM-2-XS (En)	0.3	0.2	0.5	64.3	62.2	30.6	23.9	20.8	35.9	32.1	27.2
PaLM-2-XS (Hi)	3.5	5.5	9.9	**65.4**	**63.5**	33.2	25.8	22.7	49.3	46.8	40.7
PaLM-2-XS (Lang)	**18.4**	**16.4**	**13.0**	65.1	53.3	**35.8**	**27.6**	**26.1**	**53.3**	**51.5**	**42.2**
PaLM-2-S (En)	0.4	0.2	0.5	67.4	66.8	27.5	19.9	19.9	48.6	47.1	40.8
PaLM-2-S (Hi)	4.4	6.9	13.2	68.5	**67.5**	34.2	27.0	24.9	58.3	57.0	49.0
PaLM-2-S (Lang)	**22.4**	**19.8**	**15.1**	**69.9**	57.3	**36.6**	**30.3**	**28.6**	**60.1**	**61.4**	**53.6**
PaLM-2-L (En)	0.4	0.2	0.6	71.7	69.8	37.7	33.2	29.7	28.7	27.5	26.2
PaLM-2-L (Hi)	4.7	7.0	13.8	**72.6**	**71.0**	39.7	34.6	31.2	45.5	44.8	41.5
PaLM-2-L (Lang)	**25.2**	**23.1**	**17.5**	72.5	59.8	**41.9**	**36.7**	**34.6**	**57.3**	**57.9**	**53.9**

Table 5: **Effect of in-context exemplar language** (§3.4): Performance comparison when the one-shot exemplar is provided in English (En) or Hindi (Hi) as opposed to the language of the test instance (Lang). In-context prompting in the test language (Lang) provides the best performance, followed by Hindi (Hi) and then English (En). This follows the same order as relatedness between test and prompting language, highlighting the benefit of prompting in a language more related to the test language (e.g., Hindi compared to English in this case).

	CrossSum-In			XQuAD-In		XorQA-In-Xx			XorQA-In-En		
Model	**Higher**	**Medium**	**Low**	**Higher**	**Medium**	**Higher**	**Medium**	**Low**	**Higher**	**Medium**	**Low**
mT5 models – Fine-Tuned											
mT5-B	19.5	18.9	15.1	46.2	30.9	3.8	4.0	5.5	31.7	31.4	30.8
mT5-L	20.5	19.9	15.5	54.3	38.6	11.8	11.0	10.4	56.8	53.7	45.4
mT5-XL	22.7	21.1	15.3	57.4	40.5	20.7	13.5	15.6	58.2	56.2	46.5
mT5-XXL	25.9	24.2	10.4	**62.0**	**44.4**	28.8	**23.6**	**21.9**	**70.3**	**68.9**	**59.1**
PaLM-2 models - Fine-Tuned											
PaLM-2-XXS	22.5	19.7	16.5	41.2	18.1	18.1	10.9	12.9	60.2	56.9	50.9
PaLM-2-XS	**28.5**	**25.6**	**18.8**	40.2	16.9	**30.4**	**23.6**	19.6	69.1	66.6	56.6
PaLM-2 models - Few-shot prompted											
PaLM-2-XXS$_{FS}$	7.7	7.6	6.7	37.2	26.8	22.7	12.3	16.4	51.6	47.1	38.4
PaLM-2-XS$_{FS}$	18.4	16.4	13.0	65.1	53.3	39.2	32.0	29.5	67.0	65.3	56.5

Table 6: (Top) **Fine-tuning performance** of mT5 and PaLM-2 models (§3.5). **Bold** represents best numbers among fine-tuned models. PaLM-2 outperforms mT5 for longer-form generation task (CrossSum-In), whereas mT5 models do well on short answer-span QA tasks. (Bottom) **Comparison of in-context learning vs. fine-tuning** on PaLM-2 models. In Green, we highlight the best PaLM-2 number (among fine-tuned and few-shot). For CrossSum-In task requiring longer-form generation, fine-tuning outperforms few-shot prompting.

In which requires generating a longer summary, we find that PaLM-2 models are more effective.

For Question-Answering tasks, as the model size increases from PaLM-2-XXS to PaLM-2-XS, we see that in-context learning yields equal or better performance compared to fine-tuning the model. For example, in XorQA-In-Xx, as the model size increases from XXS to XS, we see that the gap between few-shot prompting and fine-tuning significantly increases from 2-4% (in XXS) to 9-10% (in XS). In the case of XQuAD-In, we see that for the larger PaLM-2-XS model, its much better to perform in-context learning as compared to fine-tuning, for both medium and high resource Indic languages. For XorQA-In-En, in-context learning reaches the fine-tuning performance as model size increases to PaLM-2-XS. For the CrossSum-In, the gap between fine-tuning and in-context learning is reducing as model size increases, which reinforces that for even larger model sizes, it might be better to learn in-context.

public, we cannot attribute this performance difference to model sizes.

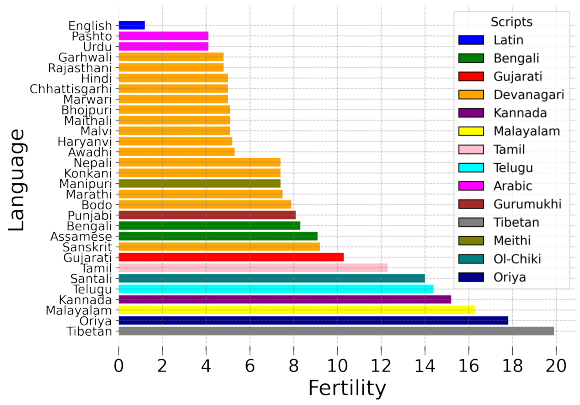

Figure 2: Tokenizer fertility for different languages using OpenAI's Byte Pair Encoding. We note that mid-low resource languages suffer from high token fertility. (Section 3.6)

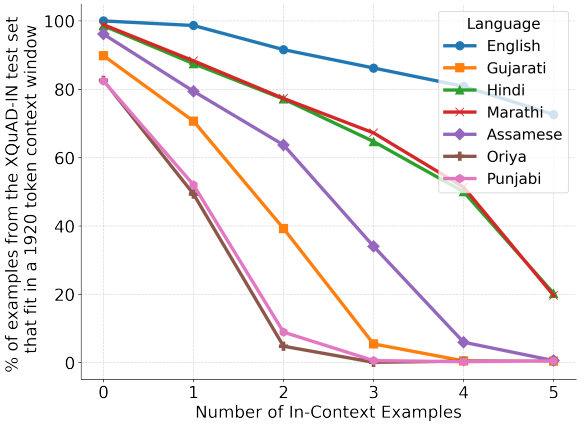

Figure 3: Percentage of the XQuAD-In test set in few-shot learning setting that fits in a 1920 token context. High token fertility of mid to low resource languages results in being able to fit much fewer in-context examples compared to higher resourced ones. (§3.6)

3.6 Analyzing Tokenizer across Indic languages

In Figure 2, we compare the token fertility (average number of sub-words that a word is broken down into by the tokenizer) across all Indic langugaes in IndicGenBench.[8] We find that the token fertility varies significantly across languages; from 4.1 for Pashto to 19.9 for Tibetan.

A high token fertility is undesirable and can disproportionately effect a particular language's performance. For languages where text is broken into more number of tokens, fewer in-context examples can be input to the LLM during inference. This can negatively impact performance (see Table 4). In Figure 3, we show how the percentage of a dataset that fits in a particular context length changes with number of in-context examples for various languages. For example, we see in Figure 3 that for medium resource languages with high token-fertility like Oriya and Punjabi we can in-corporate much fewer in-context examples compared to Indic languages with lower token-fertility like Hindi and Marathi. Analysis in this section is inspired by prior work on studying the impact of tokenization on model utility, inference, and financial cost (Ahia et al., 2023; Petrov et al., 2023). Our work supports and compliments these studies for a range of Indic languages. Alternate tokenization and encoding approaches are required to bridge the tokenization disparity between different languages. An example of this is MYTE (Limisiewicz et al., 2024), a morphology based byte encoding scheme which leads to more equitable text representations.

3.7 Qualitative Analysis

We manually analyze predictions from the best performing model PaLM-2-L with the aim to understand the shortcomings of current LLMs and highlight areas of improvements for future research. We randomly select 20 examples each in the CrossSum-In and Flores-In tasks for the following languages which are reviewed by native speakers: Awadhi, Haryanvi, Chhatisgarhi, Konkani, and Assamese. We found the following patterns of errors:

Generation in a related language The languages Awadhi, Haryanvi, and Chhatisgarhi are related to a higher resource language Hindi and written in the same script Devanagari. We find that the model generates mixed-language output with words mixed from Hindi and also outputs incorrectly inflected forms of the main verbs in the output. We show couple of examples of this phenomena in Figure 5a in the appendix.

Hallucination and Missing Information In the cross-lingual summarization task CrossSum-In, we find that the model often outputs extra information that is not present in the source article. In translation, we have observed examples where come crucial information from the source sentence is missing from the generated output. Also, in some cases, the model fails to understand polysemous English words and generates translation for the incorrect sense. We show examples of these phenomena in Figures 4a, 4b, and 5b in the appendix.

[8] We use OpenAI's BPE tokenizer (platform.openai.com/tokenizer). PaLM-2 tokenizer is not publicly available.

4 Related Work

In the last few years, many multilingual LLMs have been developed—starting from mBART (Liu et al., 2020) trained on 25 languages to LLMs that are pre-trained on hundreds of languages, such as mT5 (Xue et al., 2021), PaLM-2 (Anil et al., 2023), GPT-4 (Achiam et al., 2023), Gemini (Google, 2023), and others. These LLMs are typically evaluated on individual multilingual tasks for Translation: WMT (Farhad et al., 2021), FLORES (NLLB-Team et al., 2022); Question-Answering: XQuAD (Artetxe et al., 2020), TyDiQA (Clark et al., 2020), XorQA (Asai et al., 2021); Summarization: XLSUM (Hasan et al., 2021a); Reasoning: MGSM (Shi et al., 2022), XCOPA (Ponti et al., 2020) to name a few, or on multilingual benchmarks such as, XTREME (Hu et al., 2020) and XTREME-UP (Ruder et al., 2023). However, most of these evaluation resources contain only a handful of languages or do not contain data for low resource languages, especially Indics. Besides, cross-lingual evaluation data is even more sparse. This work is an effort to bridge these gaps by releasing IndicGenBench, a suite of datasets covering diverse cross-lingual and multilingual generation tasks in Indic languages.

Most work on creating evaluation data on Indic languages have focused on natural language understanding (NLU) tasks. Kakwani et al. (2020) and Doddapaneni et al. (2023) have released NLU test sets in Indic languages for a wide variety of tasks such as QA and NLI. Naamapadam (Mhaske et al., 2023) is a named entity recognition dataset specifically for Indic languages, MASSIVE (FitzGerald et al., 2022) is a slot-filling and intent classification dataset available in 7 Indic languages, IndicGLUE (Kakwani et al., 2020) is an NLU benchmark for 11 Indic languages, whereas GLUECoS (Khanuja et al., 2020) is a Hindi-English code-mixed benchmark, containing various NLU tasks. The Belebele Benchmark (Bandarkar et al., 2023) is a multiple-choice machine reading comprehension dataset for 122 languages of which 17 are Indic. On the other hand, IndicGenBench is a natural language generation (NLG) benchmark.

Recently, there has been work in creating evaluation benchmarks for natural language generation (NLG) on Indic languages. IndicNLG Suite (Kumar et al., 2022), consisting of 5 NLG taks in 11 Indic languages, is a leap in this direction. These datasets in this suite are automatically created, either using data from the web (e.g., Wikipedia) or using translation systems. There are few works which create evaluation data for individual tasks in Indic languages. For example, IndicTrans2 (Gala et al., 2023) creates an n-way parallel dataset for machine translation in 22 scheduled Indian Languages, Mukhyansh (Madasu et al., 2023) and PMIndiaSum (Urlana et al., 2023) are headline generation datasets for 8 and 14 Indic languages respectively, and TeSum (Urlana et al., 2022) is an abstractive summarization dataset in the Telugu language. Ramesh et al. (2022) introduced Samanantar, a large translation dataset covering 11 Indic languages. Our work complements IndicNLGSuite and the other datasets in multiple ways. IndicGenBench is manually annotated ensuring high-quality, noise-free text which is not typically found on the web. Our benchmark contains evaluation data for a much larger set of languages spanning low, medium and high resource. Our datasets are multi-language parallel enabling better comparison among different languages. Lastly, we focus on a complementary and challenging set of tasks, including cross-lingual summarization, cross-lingual and multilingual question answering, and translation.

5 Conclusion

We release IndicGenBench, the largest benchmark for evaluating LLMs on 5 user-facing generation tasks across 29 Indic languages, providing evaluation data for many under-represented Indic languages for the first time. IndicGenBench is broad coverage along many dimensions – it covers 13 writing scripts, 4 language families, and spans languages across the available web resource spectrum. We carry out extensive comparison of current SoTA LLMs on IndicGenBench and highlight areas for future improvement. We are hopeful IndicGenBench will play an important role in further development of LLMs in Indic languages ultimately benefiting a billion-plus population.

6 Limitations

Since IndicGenBench extends existing benchmarks to new Indic languages through human translation, it may miss some India-specific entities and linguistic nuances. Future work can explore translocalization for creating improved evaluation and fine-tuning. IndicGenBench doesn't cover long-form generation and reasoning tasks. Creating such datasets is part of our future work.

Acknowledgments

We thank Aditi Chaudhury, Ashok Popat, Shachi Dave, Sagar Gubbi, Megh Umekar and members of the Languages team at Google Research India (GRI) for providing feedback on this work. The authors would like to thank Manish Gupta and Divy Thakkar for their support and guidance.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David R. Mortensen, Noah A. Smith, and Yulia Tsvetkov. 2023. Do all languages cost the same? tokenization in the era of commercial language models. *ArXiv*, abs/2305.13707.

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. 2023a. Mega: Multilingual evaluation of generative ai. In *EMNLP 2023*.

Sanchit Ahuja, Divyanshu Aggarwal, Varun Gumma, Ishaan Watts, Ashutosh Sathe, Millicent Ochieng, Rishav Hada, Prachi Jain, Maxamed Axmed, Kalika Bali, and Sunayana Sitaram. 2023b. Megaverse: Benchmarking large language models across languages, modalities, models and tasks.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Akari Asai, Jungo Kasai, Jonathan Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. 2021. XOR QA: Cross-lingual open-retrieval question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 547–564, Online. Association for Computational Linguistics.

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

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2023. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants.

Ankur Bapna, Isaac Caswell, Julia Kreutzer, Orhan Firat, Daan van Esch, Aditya Siddhant, Mengmeng Niu, Pallavi N. Baljekar, Xavier García, Wolfgang Macherey, Theresa Breiner, Vera Axelrod, Jason Riesa, Yuan Cao, Mia Xu Chen, Klaus Macherey, Maxim Krikun, Pidong Wang, Alexander Gutkin, Apurva Shah, Yanping Huang, Z. Chen, Yonghui Wu, and Macduff Hughes. 2022. Building machine translation systems for the next thousand languages. *ArXiv*, abs/2205.03983.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, Yuan-Fang Li, Yong-Bin Kang, and Rifat Shahriyar. 2023. CrossSum: Beyond English-centric cross-lingual summarization for 1,500+ language pairs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2541–2564, Toronto, Canada. Association for Computational Linguistics.

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.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

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, Toronto, Canada. Association for Computational Linguistics.

Akhbardeh Farhad, Arkhangorodsky Arkady, Biesialska Magdalena, Bojar Ondřej, Chatterjee Rajen, Chaudhary Vishrav, Marta R Costa-jussa, España-Bonet Cristina, Fan Angela, Federmann Christian, et al. 2021. Findings of the 2021 conference on machine translation (wmt21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88. Association for Computational Linguistics.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. 2022. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages.

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*.

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

Gemini Team Google. 2023. Gemini: A family of highly capable multimodal models. *ArXiv*, abs/2312.11805.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021a. 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.

Tahmid Hasan, Abhik Bhattacharjee, Md Saiful Islam, Kazi Samin Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021b. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning*, pages 4411–4421. PMLR.

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

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

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. GLUECoS: An evaluation benchmark for code-switched NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3575–3585, Online. Association for Computational Linguistics.

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.

Tomasz Limisiewicz, Terra Blevins, Hila Gonen, Orevaoghene Ahia, and Luke Zettlemoyer. 2024. Myte: Morphology-driven byte encoding for better and fairer multilingual language modeling.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Lokesh Madasu, Gopichand Kanumolu, Nirmal Surange, and Manish Shrivastava. 2023. Mukhyansh: A headline generation dataset for Indic languages. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 620–634, Hong Kong, China. Association for Computational Linguistics.

Arnav Mhaske, Harshit Kedia, Sumanth Doddapaneni, Mitesh M. Khapra, Pratyush Kumar, Rudra Murthy, and Anoop Kunchukuttan. 2023. Naamapadam: A large-scale named entity annotated data for Indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10441–10456, Toronto, Canada. Association for Computational Linguistics.

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.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Poko-

rny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report.

Aleksandar Petrov, Emanuele La Malfa, Philip H. S. Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. In *Advances in Neural Information Processing Systems*.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

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 F. J. Mellor, Irina Higgins, Antonia Creswell, Nathan McAleese, Amy Wu, Erich Elsen, Siddhant M. Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, L. Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, N. K. Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Tobias 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 G. Johnson, Blake A. Hechtman, Laura Weidinger, Iason Gabriel, William S. Isaac, Edward Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem W. Ayoub, Jeff Stanway, L. L. Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. Scaling language models: Methods, analysis & insights from training gopher. *ArXiv*, abs/2112.11446.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing*.

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.

Sebastian Ruder, Jonathan H Clark, Alexander Gutkin, Mihir Kale, Min Ma, Massimo Nicosia, Shruti Rijhwani, Parker Riley, Jean-Michel A Sarr, Xinyi Wang, Nitish Gupta, et al. 2023. Xtreme-up: A user-centric scarce-data benchmark for under-represented languages. *arXiv preprint arXiv:2305.11938*.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. 2022. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Siamak Shakeri, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023. Ul2: Unifying language learning paradigms.

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.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Ashok Urlana, Pinzhen Chen, Zheng Zhao, Shay Cohen, Manish Shrivastava, and Barry Haddow. 2023. PMIndiaSum: Multilingual and cross-lingual headline summarization for languages in India. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11606–11628, Singapore. Association for Computational Linguistics.

Ashok Urlana, Nirmal Surange, Pavan Baswani, Priyanka Ravva, and Manish Shrivastava. 2022. TeSum: Human-generated abstractive summarization corpus for Telugu. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5712–5722, Marseille, France. European Language Resources Association.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

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.

Mengjie Zhao, Yi Zhu, Ehsan Shareghi, Ivan Vulić, Roi Reichart, Anna Korhonen, and Hinrich Schütze. 2021a. A closer look at few-shot crosslingual transfer: The choice of shots matters. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5751–5767, Online. Association for Computational Linguistics.

Tony Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021b. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*.

Appendix

A Resource-wise Language Classification

See Table 7 for the language resource based classification of the languages studied in our work. This classification is done on the basis of multiple criteria, such as whether google translate supports the particular language, the classification of Indic languages suggested by Doddapaneni et al. (2023) which in turn uses Joshi et al. (2020).

B Detailed and Additonal Experimental Results

In this section, we report (a) one-shot results across all models and tasks averaged over different language categories (B.1); (b) performance for different tasks and models by varying the number of in-context exemplars shown in few-shot prompting (B.2); (c) one-shot performance for the largest model in each LLM family across all languages (B.3); (d) fine-tuning performance for the largest models across all languages (B.4).

B.1 Resource wise one-shot results

In Table 10 we show performance for all LLMs considered in this paper on the one-shot prompting method across different language categories.

B.2 Performance by varying number of in-context examples

In Tables 11, 12, 13, and 14 we show additional results on how performance changes with the number of in-context examples across various tasks in IndicGenBench.

B.3 Language wise one-shot performance across models

In Tables 16, 17, 18, 19 we show language wise breakdown of performance for largest/best models – LLaMA-65B, GPT-4, PaLM-2-L in one-shot setting, across all IndicGenBench tasks.

B.4 Language wise fine-tuning performance across models

In Tables 22, 21, 20 we show language wise breakdown of performance for largest mT5 and PaLM-2 models that we fine-tune on CrossSum-In, XQuAD-In, XorQA-In-Xx, XorQA-In-En.

B.5 Indic Specific LLMs

To the best of our knowledge, there is no LLM that is pre-trained or fine-tuned on a broad set of Indic languages, or a majority of the 29 languages we study in this work. Some Indic LLMs such as OpenHathi (or its instruction-tuned variant Airavata (Gala et al., 2024)), Tamil-LLama (Balachandran, 2023), or Kan-LLama are primarily trained/fine-tuned on a single Indic language (Hindi, Telugu, and Kannada, respectively).

In this section we evaluate an Indic specific LLM, Airavata which is primarily pre-trained and fine-tuned on English and Hindi language data. We evaluate the model on 6 Indic languages: Hindi (hi), Bengali (bn), Punjabi (pa), Assamese (as), Manipuri (mni) and Santali (sat) (2 each from higher, medium and low resource), and English. See Table 8 for the results.

We find that Airavata performs reasonably well only on the Hindi language, and its performance on other Indic languages is much worse (as expected since it is only trained for Hindi). Airavata performs significantly better than LLaMA on the Hindi language, across tasks, except for CSSUM-IN summarization tasks. BLOOMZ outperforms Airavata on most tasks in the Hindi language.

C Hyperparameters

For Fine-tuning experiments on mT5, we hyperparameter search for batch size in the range {16, 32, 64} and learning rate in {5e-3, 1e-3, 5e-4, 1e-4, 5e-5, 1e-5} and select the best hyperparameters based on the Token-F1 (for Question answering datasets) score or ChrF (for CrossSum-In, Flores-In experiments) score on the validation set of the dataset. For PaLM-2 fine-tuning experiments, we hyperparameter search for batch size in the range {16, 32, 64} and learning rate in {5e-4, 1e-4, 5e-5, 1e-5} and select the best hyperparameters based on the Token-F1 score or ChrF score, as in the case of mT5 experiments. We keep temperature=0 as the default setting while decoding using PaLM-2 and GPT family of models. For LLaMA models we increased the temperature to 0.1, since otherwise, the performance of LLaMA were coming out to be close to 0 for many of our datasets. For each dataset, we fix a prompt for all models since searching for a prompt would have led to a much larger compute cost for training and evaluation. We also choose in-context exemplars randomly from the training set (from the dev set in the case of

Code	Language	Script	Family	Resource
en	English	Latin	Germanic	high
bn	Bengali	Bengali	Indo-European	higher (Indic relative)
gu	Gujarati	Gujarati	Indo-European	higher (Indic relative)
hi	Hindi	Devanagari	Indo-European	higher (Indic relative)
kn	Kannada	Kannada	Dravidian	higher (Indic relative)
ml	Malayalam	Malayalam	Dravidian	higher (Indic relative)
mr	Marathi	Devanagari	Indo-European	higher (Indic relative)
ta	Tamil	Tamil	Dravidian	higher (Indic relative)
te	Telugu	Telugu	Dravidian	higher (Indic relative)
ur	Urdu	Arabic	Indo-European	higher (Indic relative)
as	Assamese	Bengali	Indo-European	med
bho	Bhojpuri	Devanagari	Indo-European	med
ne	Nepali	Devanagari	Indo-European	med
or	Odia	Odia	Indo-European	med
pa	Punjabi	Gurumukhi	Indo-European	med
ps	Pashto	Arabic	Indo-European	med
sa	Sanskrit	Devanagari	Indo-European	med
awa	Awadhi	Devanagari	Indo-European	low
bgc	Haryanvi	Devanagari	Indo-European	low
bo	Tibetan	Tibetan	Sino-Tibetan	low
brx	Bodo	Devanagari	Sino-Tibetan	low
gbm	Garhwali	Devanagari	Indo-European	low
gom	Konkani	Devanagari	Indo-European	low
hne	Chhattisgarhi	Devanagari	Indo-European	low
hoj	Rajasthani	Devanagari	Indo-European	low
mai	Maithili	Devanagari	Indo-European	low
mni	Manipuri	Meithi	Sino-Tibetan	low
mup	Malvi	Devanagari	Indo-European	low
mwr	Marwari	Devanagari	Indo-European	low
sat	Santali	Ol Chiki	Austroasiatic	low

Table 7: Resource based language classification into relatively higher, medium and low resource for the languages studied in our work. As mentioned previously, we note that the languages classified higher, e.g., Hindi or Bengali, are in fact mid-low Web resource when compared to English and other truly high resource languages globally. For example, using Wikipedia as a proxy for language resources, compared to 6.6M+ Wikipedia articles in English, there are only 160K Hindi Wikipedia articles. See Appendix §A for more details.

	CSSUM-IN			FLORES-IN (enxx)			XQUAD-IN			XORQA-IN-XX			XORQA-IN-EN		
Language	Airavata	LLaMA	BLOOMZ	Airavata	LLaMA	BLOOMZ	Airavata	LLaMA	BLOOMZ	Airavata	LLaMA	BLOOMZ	Airavata	LLaMA	BLOOMZ
en	31.6	19.1	13.9	-	-	-	75.7	49.2	86.8	69.5	56.0	72.2	69.5	59.7	68.7
hi	0.3	6.0	1.1	55.9	19.0	66.8	48.7	16.6	64.3	46.8	30.3	18.6	31.6	19.4	67.6
bn	0.0	2.0	0.2	8.5	9.6	73.9	6.5	2.5	57.9	0.2	3.8	0.3	33.8	12.6	70.0
pa	0.1	1.2	0.3	6.8	6.8	55.7	3.2	0.8	60.2	2.5	0.9	4.7	21.2	1.8	67.3
as	0.3	2.0	1.4	5.8	5.5	50.1	5.2	2.1	52.8	0.5	3.0	2.6	32.4	9.4	57.7
mni	0.2	2.3	0.2	6.1	6.7	6.5	-	-	-	1.4	2.9	0.3	22.6	5.4	21.5
sat	0.0	2.4	0.3	6.4	8.3	15.9	-	-	-	1.1	1.1	0.0	18.6	5.7	5.5

Table 8: One-Shot performance on INDICGENBENCH comparing language-specific Indic LLM Airavata with other similar sized open-source models. Airavata is primarily pre-trained and fine-tuned for Hindi and English. As expected, Airavata outperforms other LLMs on Hindi whereas it's performance on other Indic languages is significantly worse than broad-coverage LLMs like BLOOMZ. See Appendix §B.5 for more details.

Code	Language	Coverage in IndicGenBench				
		CrossSum-In	Flores-In	XQuAD-In	XorQA-In-Xx	XorQA-In-En
bn	Bengali	✓	✓	✓	✓	✓
gu	Gujarati	✓	✓	✓	✓	✓
hi	Hindi	✓	✓	✓	✓	✓
kn	Kannada	✓	✓	✓	✓	✓
ml	Malayalam	✓	✓	✓	✓	✓
mr	Marathi	✓	✓	✓	✓	✓
ta	Tamil	✓	✓	✓	✓	✓
te	Telugu	✓	✓	✓	✓	✓
ur	Urdu	✓	✓	✓	✓	✓
as	Assamese	✓	✓	✓	✓	✓
bho	Bhojpuri	✓	✓		✓	✓
ne	Nepali	✓	✓			
or	Odia	✓	✓	✓	✓	✓
pa	Punjabi	✓	✓	✓	✓	✓
ps	Pashto	✓	✓		✓	✓
sa	Sanskrit	✓	✓		✓	✓
awa	Awadhi	✓	✓		✓	✓
bgc	Haryanvi	✓	✓		✓	✓
bo	Tibetan	✓	✓		✓	✓
brx	Bodo	✓	✓		✓	✓
gbm	Garhwali	✓	✓		✓	✓
gom	Konkani	✓	✓		✓	✓
hne	Chhattisgarhi	✓	✓		✓	✓
hoj	Rajasthani	✓	✓		✓	✓
mai	Maithili	✓	✓		✓	✓
mni	Manipuri	✓	✓		✓	✓
mup	Malvi	✓	✓		✓	✓
mwr	Marwari	✓	✓		✓	✓
sat	Santali	✓	✓		✓	✓

Table 9: Coverage of languages across different tasks in IndicGenBench.

Flores-In). This decision is a limitation of this work, since it has been shown that prompt and in-context exemplars can significantly impact an LLMs performance (Zhao et al., 2021b,a). The prompts we use are provided in §F. Searching for better prompts and exemplars is left as future work. For all fine-tuning runs, we just perform one run (as opposed to taking average of multiple runs) due to prohibitive costs of fine-tuning LLMs multiple times.

D Computing Infra

For mT5 fine-tuning runs, we use a combination of 4 to 64 TPU-V3, and for PaLM-2 we use up to 256 TPU-V4. All experiments take about 6 hours to 1-2 days of time depending upon the resources and size of the model being trained. We use NVIDIA-V100 16GB GPUs for evaluating open-source models like BLOOM, BLOOMZ, Airavata on IndicGenBench.

E Dataset Licenses

In compliance with licenses of the original datasets, we release our evaluation datasets under the following licenses: (a) XQuAD-In and Flores-In under the CC BY-SA 4.0 license; (b) CrossSum-In under the CC BY-NC-SA 4.0 license; and (d) XorQA-In under the MIT license.

	XorQA-In-En			XQuAD-In		CrossSum-In			XorQA-In-Xx			Flores-In		
Model	**High**	**Medium**	**Low**	**High**	**Medium**	**High**	**Medium**	**Low**	**High**	**Medium**	**Low**	**High**	**Medium**	**Low**
LLaMA-7B	9.9	7.6	12.2	4.7	1.1	3.6	3.6	3.7	8.1	3.4	8.8	11.2/20.4	9.6/20.2	12.8/23.2
LLaMA-13B	10.5	9.9	14.2	5.5	1.5	3.9	4.1	4.2	9.6	6.3	12.8	12.6/22.5	11.3/22.0	14.8/26.4
LLaMA-65B	16.4	14.0	17.3	8.8	1.9	4.4	4.6	4.7	17.7	13.5	17.1	18.2/31.5	15.4/30.0	19.5/35.0
Gemma-7B-PT	25.7	22.5	23.1	0.6	0.1	0.0	0.0	0.0	16.5	6.9	10.5	40.7 / 58.8	25.5 / 49.5	29.8 / 45.0
Gemma-7B-IT	29.5	23.9	21.9	38.8	24.8	13.9	11.5	10.0	18.9	8.3	12.2	17.6 / 33.7	15.0 / 26.1	21.3 / 27.7
BLOOM-7B	29.3	21.6	20.6	15.2	9.8	3.1	3.8	4.3	10.8	5.0	7.2	19.8 / 34.7	16.3 / 31.4	18.4 / 28.6
BLOOMZ-7B	**64.7**	45.8	39.5	55.5	48.1	1.5	1.7	0.6	10.8	2.8	6.2	**67.7** / 59.1	39.4 / 50.2	22.9 / 40.0
GPT-3.5	39.0	34.9	33.4	36.2	23.9	17.6	16.5	15.3	24.2	20.0	20.6	32.9/52.9	27.3/48.2	27.6/43.8
GPT-4	49.4	50.0	41.8	56.1	54.6	19.4	17.9	16.3	25.8	21.6	22.6	36.2/59.6	30.7/55.2	29.9/50.5
PaLM-2-XXS	43.6	38.3	31.5	37.2	26.8	7.7	7.6	6.7	17.7	8.8	12.8	31.7 / 52.1	19.0 / 43.2	21.4 / 37.6
PaLM-2-XS	53.3	51.5	42.2	65.1	53.3	18.4	16.4	13.0	35.8	27.6	26.1	52.0 / 64.8	39.0 / 60.6	33.7 / 52.6
PaLM-2-S	60.1	**61.4**	53.6	69.9	57.3	22.4	19.8	15.1	36.6	30.3	28.6	54.7 / 66.8	42.5 / 63.5	36.4 / 57.0
PaLM-2-L	57.3	57.9	**53.9**	**72.5**	**59.8**	**25.2**	**23.1**	**17.5**	**41.9**	**36.7**	**34.6**	56.9 / **68.2**	**45.9 / 65.6**	**41.9 / 62.6**

Table 10: **One-Shot performance on IndicGenBench**. Results are averaged over high, medium and low resource language examples present in the dataset. For question answering we measure the F1/EM score while for summarization and translation we measure ChrF score. See Appendix §B.1 for more details.

	Shots	
Model (LLM)	**0**	**1**
LLaMA-7B	2.4	3.7
LLaMA-13B	1.5	4.1
LLaMA-65B	5.1	4.6
PaLM-2-XXS	0.2	7.2
PaLM-2-XS	1.0	15.5
PaLM-2-S	4.1	18.5
PaLM-2-L	**7.7**	**21.2**

Table 11: Few-shot results for PaLM-2 models on CrossSum-In dataset. ChrF scores are reported.

	Shots	
Model (LLM)	**0**	**1**
LLaMA-7B	4.3	3.8
LLaMA-13B	4.7	4.5
LLaMA-65B	7.2	7.1
PaLM-2-XXS	19.0	34.6
PaLM-2-XS	38.7	62.2
PaLM-2-S	44.7	66.7
PaLM-2-L	**50.6**	**69.3**

Table 12: Few-shot results for PaLM-2 models on XQuAD-In dataset. F1 scores are reported.

F Prompts

The set of prompts used for evaluations on all datasets are shown in Table 15.

	Shots			
Model (LLM)	**0**	**1**	**2**	**3**
LLaMA-7B	4.7	10.4	12.5	14.1
LLaMA-13B	2.7	12.1	15.4	17.0
LLaMA-65B	8.2	16.3	18.8	19.8
PaLM-2-XXS	31.0	36.8	41.6	44.5
PaLM-2-XS	15.6	47.8	57.8	61.7
PaLM-2-S	**40.6**	**57.4**	63.4	66.3
PaLM-2-L	31.4	55.9	**66.3**	**70.3**

Table 13: Few-shot results for PaLM-2 models on XorQA-In-En dataset. F1 scores are reported.

	# of in-context examples		
Model (LLM)	**0**	**1**	**5**
LLaMA-7B	8.0 / 10.4	11.5 / 21.6	11.4 / 19.8
LLaMA-13B	8.6 / 20.6	13.3 / 24.1	13.4 / 22.7
LLaMA-65B	14.0 / 28.0	18.1 / 32.7	18.3 / 30.3
PaLM-2-XXS	0.8 / 22.4	24.0 / 43.4	26.9 / 44.8
PaLM-2-XS	20.1 / 53.3	40.7 / 58.3	42.3 / 58.8
PaLM-2-S	24.9 / 60.1	43.5 / 61.6	45.2 / 61.9
PaLM-2-L	**31.1 / 62.2**	**47.5 / 65.1**	**49.3 / 65.7**

Table 14: Few-shot results for LLaMA and PaLM-2 models on Flores-In (both enxx / xxen direction).

Summarize the following article in **Haryanvi**:
The White Garden, at Kensington Palace, was planted to mark 20 years since Princess Diana died in a car crash. The Duchess of Cambridge joined the princes on the garden tour. A spokeswoman for Kensington Palace said: "The engagement will allow the princes to pay tribute to the life and work of their mother." They met representatives from the causes and charities supported by Diana, including the Royal Marsden and Great Ormond Street hospitals, the National Aids Trust, Centrepoint youth homelessness charity and the Leprosy Mission. Members of the public have been leaving tributes and flowers at the gates of the palace to mark the anniversary of Diana's death. The Princess of Wales died on 31 August 1997 in Paris, when William, now the Duke of Cambridge, was 15 and his brother was 12. The garden at their mother's former home has been inspired by memories of her life, style and image, such as her white "Elvis" Catherine Walker dress. The White Garden, as it is known, follows a tradition first established at Sissinghurst Castle in Kent, famous for its own white garden created in the 1930s. Their Royal Highnesses met gardener Sean Harkin who designed the display and Graham Dillamore who knew the princess when he worked there some 30 years ago. The garden has been open since spring and will continue into September with white roses, lilies, gladioli and cosmos. It is the fourth London memorial created in tribute to Diana - the others are the Diana Memorial Playground at Kensington Palace, the Diana Memorial Fountain in Hyde Park, and the Diana Memorial Walk at St James's Palace.

Model Output: प्रिंस विलियम और प्रिंस हैरी नै कैंसिंगटन पैलेस मै एक नया गार्डन खोलया सै जो उनकी माँ, प्रिंसेस डायना की या
(English translation: Prince William and Prince Harry have opened a new garden at Kensington Palace in memory of their mother, Princess Diana.)

(a) Model hallucinates that Prince William and Harry opened a new garden.

Summarize the following article in **Awadhi**:
The surgery, which lasted seven hours, took place on 14 February at the Nair hospital in the western city of Mumbai. But the procedure was not made public as doctors were not sure until now if it had been successful. "Now it's a matter of recovery but he's out of danger", Dr Trimurti Nadkarni, chief of neurosurgery, told the BBC. Santlal Pal, a shopkeeper from the northern state of Uttar Pradesh, had been living with the tumour for three years before his surgery was carried out. Doctors said Mr Pal had lost his vision because of the tumour but hoped he would regain it as he continued to recover from the operation. His wife told the Hindu newspaper that they had been told by doctors at three different hospitals in Uttar Pradesh that the tumour was inoperable. "Such cases are critical", said Dr Nadkarni, adding that Mr Pal needed 11 units of blood during the operation and after that, he was connected to a ventilator for a few days after the surgery.
Model Output: एकु 31 साल के आदमी का नाक ते 1.8 किलोग्राम का ट्यूमर न
(English translation: A 1.8 kg tumor was removed from the nose of a 31 year old man)

(b) Model hallucinates the age of the person in the article.

Figure 4: Example predictions from PaLM-2-L model on the CROSSSUM-IN task highlighting issues with hallucinations in model predictions.

Translate the following in **Awadhi**:
Input: One can only wonder what the keyboard will become when something newer comes along.
Model output: कोई केवल इ सोच सकत है कि जब कुछ नया आवत है त कीबोर्ड का होई।
Ground Truth: इ सिर्फ़ सोचि जाय सकत है कि जब कुछ नया आयी त कीबोर्ड का होई।

Translate the following in **Chhatisgarhi**:
Input: Scaffolds can be both virtual and real, in other words, a teacher is a form of scaffold but so is the little paperclip man in Microsoft Office.
Model output: मचान आभासी अउ वास्तविक दुनो हो सकत हंवय, दुसर शब्दमन में, एक शिक्षक मचान के एक रूप हवय लेकिन माइक्रोसॉफ्ट ऑफिस में छोटा पेपरक्लिप आदमी भी हवय।
Ground Truth: मचान आभासी अउ वास्तविक दूनों हो सकथे, दूसरे सब्द मन म, एकठन सिक्छ्क मचान के एक रूप हवय लेकिन माइक्रोसॉफ्ट ऑफिस म छोटा पेपरक्लिप आदमी घलो अइसा ही हवय।

(a) In these examples of translating into Awadhi and Chhatisgarhi, both low resource languages similar to a higher resource language Hindi, we find that many Hindi words and incorrect inflections for verbs are produced.

Translate the following in **Konkani**:
Input:When asked for comment, Miller said, "Mike talks a lot during the hearing...I was getting ready so I wasn't really hearing what he was saying."
Model output: टिप्पणी करपाक विचारलें तेन्ना, मिलर म्हणाले, "माईक आयकुपाच्या वेळार खूब उलोवपाक लागता ... हांव तयार जातालो म्हणून तो कितें म्हणटालो हांव खरेंच आयकनाशिल्लो"
Ground Truth: टिप्पणी मागली तेन्ना मिलरान म्हणलें, "मायक सुनावणी वेळार खूब उलयता... हांव तयार जातालो म्हणून तो कितें सांगता तें म्हाका खरेंच आयकूंक नाशिल्लें"

Translate the following in **Assamese**:
Input: Casablanca is one of the least interesting places to shop in all of Morocco.
Model output: কেচাব্লাংকা হৈছে মৰক্কোৰ সকলো ঠাইৰ ভিতৰত আটাইতকৈ কম আকৰ্ষণীয় ঠাই।
Ground Truth: মৰক্কোত কাছাবলাংকা হৈছে কিনা-কটা কৰাৰ বাবে আটাইতকৈ আকৰ্ষণহীন ঠাই।

(b) 1. In the first example of translation into Konkani, the model does not understand that the first instance of "hearing" refers to a court hearing and the second instance refers to "listening". The model incorrectly outputs words related to the "listening" sense in both instances. 2. In the translation into Assamese example, the model does not produce translation for the crucial information "to shop".

Figure 5: Example predictions from PaLM-2-L model on the FLORES-IN task highlighting issues of (a) producing words in the wrong (higher-resourced) language or words with the wrong inflection, and (b) outputting the incorrect translations for polysemous words in English or missing crucial information from the generated translation.

Dataset	Prompt
CROSSSUM-IN	I will first show a news article in English and then provide a summary of it in the [Target Language Name] language. **Summarize the following article:** [Article] **Summary:**
FLORES-IN, xxen	Translate the following: **To English:** [Sentence in Target Language] **Output:**
FLORES-IN, enxx	Translate the following: **To** [Target Language Name]: [Sentence in English] **Output:**
XQUAD-IN	[Passage in Target Language] **Q:** [Question in Target Language] **A:**
XORQA-IN-XX	Generate an answer in [Target Language Name] for the question based on the given passage: [Passage in English] **Q:** [Question in Target Language] **A:**
XORQA-IN-XX	Generate an answer in English for the question based on the given passage: [Passage in English] **Q:** [Question in Target Language] **A:**

Table 15: Prompt templates used for different datasets in INDICGENBENCH. n-shot examples have the same format as the last, test example given to the model, which comes after the n-shot examples.

Lang. Code	CROSSSUM-IN				
	LLaMA-65B	**Gemma-7B-IT**	**BLOOMZ-7B**	**GPT-4**	**PaLM-2-L**
en	26.1	24.8	18.6	30.3	41.1
bn	2.7	13.6	0.4	20.9	23.4
gu	2.1	10.3	1.0	16.5	19.2
hi	7.8	16.8	0.9	23.1	30.4
kn	2.9	13.6	0.9	15.1	25.4
ml	4.3	11.4	0.5	14.4	23.3
mr	4.1	13.6	0.6	22.4	25.4
ta	5.4	20.7	2.1	20.4	28.9
te	2.5	13.6	0.7	16.6	21.9
ur	7.5	13.0	6.4	24.9	28.9
as	2.7	12.5	1.8	17.3	24.3
bho	5.5	13.2	0.9	19.3	20.2
ne	7.4	16.8	2.0	23.2	29.1
or	2.5	4.3	0.3	10.7	23.4
pa	1.7	10.7	0.2	15.8	19.1
ps	7.0	9.2	6.3	19.9	25.2
sa	5.3	13.5	0.5	19.4	20.1
awa	5.1	12.8	0.6	18.2	19.4
bgc	5.3	12.5	0.7	17.9	20.2
bo	1.5	2.8	0.1	8.5	10.0
brx	4.9	2.8	0.4	15.5	11.0
gbm	5.8	11.4	2.0	16.2	16.2
gom	5.8	12.6	0.8	20.3	23.1
hne	5.5	13.4	0.9	19.2	22.0
hoj	5.2	12.5	0.3	17.7	17.7
mai	4.9	12.6	0.5	18.2	21.1
mni	3.3	4.4	0.2	13.2	15.6
mup	5.3	13.1	0.6	20.0	23.8
mwr	5.1	13.2	0.8	18.4	20.2
sat	2.7	5.5	0.2	8.5	7.4
Avg.	4.6	11.6	1.2	17.6	21.2
High Avg.	4.4	13.9	1.5	19.4	25.2
Medium Avg.	4.6	11.5	1.7	17.9	23.1
Low Avg.	4.7	10.0	0.6	16.3	17.5

Table 16: Performance comparison of models on CROSSSUM-IN in one-shot setting across all supported languages.

Lang. Code	FLORES-IN (enxx)					FLORES-IN (xxen)				
	LLaMA-65B	**Gemma-7B-IT**	**BLOOMZ-7B**	**GPT-4**	**PaLM-2-L**	**LLaMA-65B**	**Gemma-7B-IT**	**BLOOMZ-7B**	**GPT-4**	**PaLM-2-L**
bn	18.7	27.3	71.9	40.0	54.0	37.4	35.1	60.4	59.9	66.8
gu	11.3	3.3	68.1	31.3	56.1	22.1	29.1	60.9	61.2	70.9
hi	33.2	35.7	65.2	48.6	60.2	53.7	48.4	57.3	65.2	70.5
kn	11.1	15.6	66.8	29.9	58.0	20.6	27.6	59.1	57.6	65.7
ml	14.5	7.1	66.2	28.4	59.7	22.8	28.8	60.6	58.6	68.0
mr	22.7	22.3	68.6	39.2	53.1	38.2	31.7	58.8	59.8	68.5
ta	16.8	29.8	75.2	34.5	60.0	24.5	30.5	55.7	53.9	65.7
te	10.7	16.1	68.8	30.7	60.5	21.7	33.0	56.1	58.1	70.5
ur	24.6	1.3	58.5	43.5	50.6	42.8	38.8	62.8	62.0	67.2
awa	26.4	29.1	32.1	38.8	49.9	46.1	36.9	56.3	61.8	69.3
bgc	26.6	28.9	31.1	38.0	49.8	45.1	33.3	55.7	63.5	72.1
bo	6.6	4.5	13.7	17.0	40.9	16.9	17.0	16.1	27.7	50.1
brx	12.4	13.0	12.1	17.2	11.3	20.7	17.8	17.0	29.2	31.0
gbm	24.0	26.6	28.3	36.2	47.9	43.0	31.2	49.6	62.2	72.1
gom	17.8	19.0	17.3	27.5	39.0	29.7	23.3	28.3	48.2	63.2
hne	25.4	29.2	29.3	38.4	52.7	42.8	32.9	53.6	62.3	75.1
hoj	23.7	26.6	28.1	36.0	48.1	41.9	30.4	51.3	62.3	73.0
mai	21.4	26.0	22.8	36.2	54.1	41.8	32.7	57.5	60.9	73.3
mni	8.3	9.3	8.7	15.2	19.5	20.7	18.1	18.7	31.8	41.7
mup	27.0	28.3	31.7	39.9	51.9	45.1	34.9	49.2	62.3	73.0
mwr	26.4	29.7	33.9	39.2	52.2	44.6	34.6	51.3	64.2	74.5
sat	8.1	6.5	8.4	9.4	27.9	16.4	17.0	16.1	20.6	45.6
as	9.5	18.1	51.1	28.5	44.1	25.9	23.6	60.0	51.7	63.3
bho	23.4	25.1	24.9	33.5	42.9	40.9	32.0	51.8	55.3	61.7
ne	25.9	24.7	72.0	45.8	57.8	42.3	34.6	63.3	62.6	72.3
or	9.4	5.2	45.7	19.6	52.9	19.6	17.1	63.4	56.9	68.0
pa	9.7	2.6	57.1	30.7	51.8	22.1	28.4	60.8	62.5	70.1
ps	13.3	10.9	9.2	27.3	37.6	27.7	22.0	19.4	45.7	64.5
sa	16.9	18.6	15.7	29.3	34.3	31.4	25.1	32.5	51.5	59.4
Avg.	18.1	18.6	40.8	32.1	47.5	32.7	29.2	48.4	54.5	65.1
High Avg.	18.2	17.6	67.7	36.2	56.9	31.5	33.7	59.1	59.6	68.2
Medium Avg.	15.4	15.0	39.4	30.7	45.9	30.0	26.1	50.2	55.2	65.6
Low Avg.	19.5	21.3	22.9	29.9	41.9	35.0	27.7	40.0	50.5	62.6

Table 17: Performance comparison of models on FLORES-IN in one-shot setting across all supported languages.

Lang. Code	XorQA-In-Xx					XorQA-In-En				
	LLaMA-65B	**Gemma-7B-IT**	**BLOOMZ-7B**	**GPT-4**	**PaLM-2-L**	**LLaMA-65B**	**Gemma-7B-IT**	**BLOOMZ-7B**	**GPT-4**	**PaLM-2-L**
en	61.1	35.5	68.7	37.4	71.4	63.9	34.7	69.7	37.9	71.5
bn	15.8	9.4	0.6	16.5	40.4	16.9	25.7	68.4	46.2	55.6
gu	8.8	18.3	11.2	23.8	40.2	12.9	32.7	64.2	51.9	53.7
hi	43.5	29.6	16.5	38.3	56.5	23.1	31.1	66.1	41.6	58.8
kn	8.7	19.4	8.8	28.8	41.1	9.4	26.2	63.5	54.6	59.6
ml	13.9	26.7	19.6	31.1	56.7	16.9	30.7	66.4	55.3	54.8
mr	28.1	26.4	17.7	30.4	44.5	22.3	29.1	64.0	47.9	57.4
ta	16.0	17.5	10.7	25.4	40.2	19.0	26.8	64.0	50.1	57.7
te	4.0	15.3	11.4	18.5	25.6	6.0	29.3	67.6	52.0	56.3
ur	20.6	7.6	1.1	18.9	31.6	21.5	34.1	58.5	44.9	61.5
as	17.6	10.8	2.0	26.4	45.5	12.5	22.3	60.0	51.6	59.3
bho	19.9	9.9	2.0	23.8	32.1	21.4	26.7	55.2	46.9	59.4
or	4.0	1.6	6.2	22.1	35.9	7.7	14.8	48.0	54.6	52.6
pa	6.9	13.3	4.0	25.7	40.9	10.7	30.4	63.2	47.3	55.5
ps	10.8	3.4	0.5	12.4	25.7	12.4	23.9	3.5	44.7	62.9
sa	21.9	11.0	2.1	19.0	40.3	19.0	25.5	44.8	54.8	57.8
awa	26.3	13.1	7.8	28.5	41.3	23.3	25.3	58.1	50.3	58.4
bgc	18.5	16.4	10.2	20.1	28.1	23.1	25.6	55.7	43.8	58.4
bo	3.3	0.9	5.4	32.5	32.5	3.2	14.4	6.0	33.8	52.6
brx	5.3	12.7	0.9	15.5	20.0	12.1	15.6	7.6	35.9	39.1
gbm	13.7	14.9	9.5	18.5	35.4	21.6	24.9	50.7	46.1	60.2
gom	15.1	8.0	0.8	19.2	33.2	14.0	22.1	37.8	41.4	54.2
hne	29.7	22.7	12.6	32.0	43.4	22.5	24.9	56.4	43.9	54.8
hoj	31.2	25.3	17.3	36.3	53.8	21.5	23.9	51.5	48.8	62.6
mai	28.0	18.9	8.0	29.4	45.1	20.5	25.9	55.2	45.0	55.1
mni	8.9	4.0	0.4	13.2	23.6	9.7	14.9	17.7	36.2	47.5
mup	17.0	10.6	4.8	21.0	26.4	22.8	24.9	57.2	41.9	60.6
mwr	24.4	11.1	2.9	25.3	40.0	23.2	25.8	53.1	48.7	59.1
sat	1.3	0.3	0.0	2.6	26.6	6.7	16.7	6.2	27.5	38.6
Avg.	16.5	13.5	7.0	23.4	37.4	16.3	24.8	49.0	46.0	55.9
High Avg.	17.7	18.9	10.8	25.8	41.9	16.4	29.5	64.7	49.4	57.3
Medium Avg.	13.5	8.3	2.8	21.6	36.7	14.0	23.9	45.8	50.0	57.9
Low Avg.	17.1	12.2	6.2	22.6	34.6	17.3	21.9	39.5	41.8	53.9

Table 18: Performance comparison of models on XorQA-In-Xx and XorQA-In-En in one-shot setting across all supported languages.

	XQuAD-In				
Lang. Code	**LLaMA-65B**	**Gemma-7B-IT**	**BLOOMZ-7B**	**GPT-4**	**PaLM-2-L**
en	62.0	45.7	86.7	64.8	83.7
bn	7.4	35.7	57.1	56.5	72.3
gu	0.5	40.6	57.0	53.9	72.1
hi	25.3	49.4	63.7	63.1	76.7
kn	0.4	41.1	52.0	55.1	74.4
ml	3.6	33.7	48.8	56.5	66.6
mr	14.7	33.8	58.5	57.3	76.9
ta	4.0	42.1	55.0	55.3	75.1
te	0.2	36.8	51.9	48.8	68.0
ur	22.9	35.9	55.9	58.4	70.2
as	3.1	28.4	48.8	53.0	66.6
or	0.7	5.9	34.0	51.1	52.0
pa	1.8	40.2	61.4	59.7	60.7
Avg.	7.1	35.3	53.7	55.7	69.3
High Avg.	8.8	38.8	55.5	56.1	72.5
Medium Avg.	1.9	24.8	48.1	54.6	59.8

Table 19: Performance comparison of models on XQuAD-In in one-shot setting across all supported languages.

Lang. Code	XQuAD-In	
	mT5 XXL	**PaLM-2-XS**
en	78.1	64.4
bn	53.2	47.7
gu	53.4	21.1
hi	59.2	59.1
kn	60.2	27.9
ml	52.6	30.1
mr	60.9	51.6
ta	62.8	45.1
te	51.0	28.2
ur	63.2	50.9
as	45.0	35.1
or	25.0	6.7
pa	51.3	8.8
Avg.	53.2	34.4
High Avg.	57.4	40.2
Medium Avg.	40.5	16.9

Table 20: Performance comparison of **fine-tuned** models on XQuAD-In dataset across all supported languages.

Lang. Code	CrossSum-In	
	mT5 XXL	**PaLM-2-XS**
en	31.8	36.6
bn	25.4	28.8
gu	22.4	24.6
hi	26.2	30.5
kn	26.7	28.5
ml	26.4	26.7
mr	24.4	29.8
ta	31.6	32.3
te	25.8	26.9
ur	26.5	28.6
as	23.9	25.8
bho	22.0	24.1
ne	28.3	30.5
or	23.2	24.1
ps	26.9	24.9
pa	23.4	23.7
sa	25.4	26.0
awa	20.4	21.6
brx	19.7	12.6
hne	22.5	23.9
gbm	18.3	18.1
bgc	22.0	23.0
gom	25.1	25.8
mai	22.1	23.8
mup	24.5	25.6
mni	15.6	12.0
mwr	21.0	22.2
hoj	19.5	21.2
sat	5.9	10.0
bo	7.0	5.1
Avg.	22.5	23.5
High Avg.	26.2	28.5
Medium Avg.	24.7	25.6
Low Avg.	18.7	18.8

Table 21: Performance comparison of **fine-tuned** models on CrossSum-In dataset across all supported languages.

Lang. Code	XORQA-IN-XX		XORQA-IN-EN	
	mT5 XXL	**PaLM-2-XS**	**mT5 XXL**	**PaLM-2-XS**
en	72.4	55.8	70.7	68.1
bn	15.4	16.3	71.9	68.6
gu	28.7	25.9	70.6	69.0
hi	38.3	45.8	70.6	68.5
kn	28.2	32.0	69.3	69.3
ml	38.2	44.0	71.1	70.5
mr	32.8	32.9	70.5	68.9
ta	27.2	33.8	69.0	69.8
te	21.5	24.1	70.5	70.4
ur	28.3	18.6	68.8	66.5
as	24.8	27.6	70.3	67.7
bho	20.6	22.0	67.9	66.0
or	23.9	28.6	69.1	65.3
pa	27.3	28.5	69.0	68.5
ps	22.4	13.1	68.3	64.2
sa	22.1	21.5	69.1	68.0
awa	24.2	27.7	68.0	65.2
bgc	21.1	21.5	69.1	62.5
bo	41.5	6.4	42.9	56.7
brx	14.9	7.9	38.8	30.2
gbm	15.6	18.2	65.2	62.5
gom	17.8	21.0	64.7	64.1
hne	27.9	32.4	68.2	64.7
hoj	30.8	33.3	66.2	62.1
mai	23.5	31.8	69.2	65.0
mni	16.8	8.9	48.6	37.4
mup	20.1	19.1	67.1	64.2
mwr	27.2	23.7	68.4	64.2
sat	2.7	3.3	32.1	36.5
Avg.	24.5	23.9	64.8	62.7
High Avg.	28.8	30.4	70.3	69.1
Medium Avg.	23.6	23.6	68.9	66.6
Low Avg.	21.9	19.6	59.1	56.6

Table 22: Performance comparison of **fine-tuned** models on XORQA-IN-XX and XORQA-IN-EN dataset across all supported languages.