

# INDIC QA BENCHMARK: A Multilingual Benchmark to Evaluate Question Answering capability of LLMs for Indic Languages

Abhishek kumar singh<sup>◇</sup>, Vishwajeet Kumar<sup>§</sup>, Rudra Murthy<sup>§</sup>  
Jaydeep Sen<sup>§</sup>, Ashish Mittal<sup>§</sup>, Ganesh Ramakrishnan<sup>◇</sup>

<sup>◇</sup>Indian Institute of Technology Bombay, India

<sup>§</sup>IBM Research, India

{abhisheksingh, ganesh}@cse.iitb.ac.in,

{vishk024, rmurthyv, jaydesen, arakeshk}@in.ibm.com

## Abstract

Large Language Models (LLMs) perform well on unseen tasks in English, but their abilities in non-English languages are less explored due to limited benchmarks and training data. To bridge this gap, we introduce the Indic-QA Benchmark, a large dataset for context-grounded question answering in 11 major Indian languages, covering both extractive and abstractive tasks. Evaluations of multilingual LLMs, including instruction fine-tuned versions, revealed weak performance in low-resource languages due to a strong English-language bias in their training data. We also investigated the Translate-Test paradigm, where inputs are translated to English for processing and the results are translated back into the source language for output. This approach outperformed multilingual LLMs, particularly in low-resource settings. By releasing Indic-QA, we aim to promote further research into LLMs' question-answering capabilities in low-resource languages. This benchmark offers a critical resource to address existing limitations and foster multilingual understanding.<sup>1</sup>

India, with a population of almost 1.4 billion people, is home to numerous major languages that are considered low-resource by the natural language processing (NLP) community. Despite the growing capabilities of Large Language Models (LLMs) in tasks like context-grounded question answering (CQA) in English, their performance in non-English languages remains underexplored due to a lack of high-quality datasets. To address this gap, we introduce Indic-QA, the largest publicly available context-grounded question-answering dataset for 11 major Indian languages from two language families. This dataset encompasses both extractive and abstractive QA tasks, incorporating existing datasets as well as English QA datasets translated

<sup>1</sup>Source code and Data are available at <https://github.com/ayushayush591/IndicQA-Benchmark>.

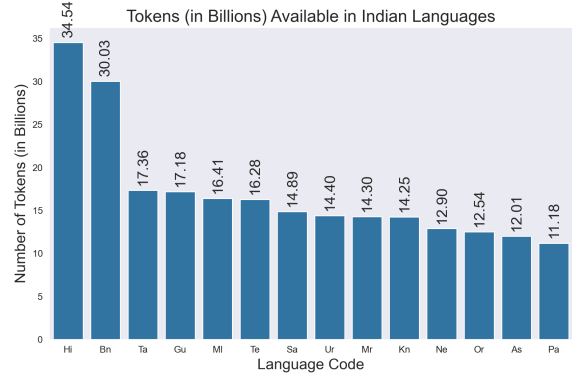


Figure 1: Total tokens available for each Indian language in the Sangraha Data (Rahman Khan et al., 2024). In contrast, RefinedWeb (Penedo et al., 2023) contains around 5 Trillion tokens in English.

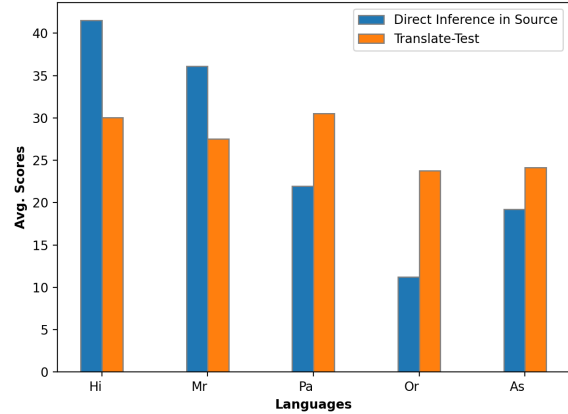


Figure 2: Comparison of **LLama 3-8B** evaluation results using source language test set vs Translate test set. The results clearly indicate that the Translate-Test paradigm yields better scores for low-resource languages (Punjabi, Odia, Assamese), whereas the source language test set gets better scores for mid-resource language Hindi and Marathi which has high correlation with Hindi.

into Indian languages. Additionally, we generate a synthetic dataset using the Gemini model, with manual verification for quality assurance. Our evaluation of various multilingual LLMs and their

instruction-fine-tuned variants on the Indic-QA benchmark reveals subpar performance, particularly for low-resource languages. This outcome highlights the English language bias inherent in these models due to predominantly English pre-training data.

We tested the Translate-Test paradigm as an alternative, which translates input from the source language to English, utilizes the LLM’s problem-solving ability in English, and then translates the response back to the source language. Our investigation shows that while multilingual LLMs perform better in mid-resource languages, the Translate-Test paradigm significantly outperforms them in low-resource languages.

**What distinguishes our benchmark from other existing Multilingual Indic context-grounded question-answering Benchmarks?** There are numerous context-grounded question-answering benchmarks available for high-resource languages like English. However, there are very few benchmarks available for Indic languages[1], and those that do exist often lack domain diversity and are limited in size. To address these gaps, we developed the Indic-QA benchmark. We sampled a variety of Wikipedia and Common Crawl pages, focusing on paragraphs rich in cultural nuances, to create a comprehensive and culturally diverse benchmark for Indic languages. Our findings show that base pre-trained models frequently produce incorrect or illogical answers. However, few-shot prompting notably improves answer quality by guiding the models to extract more precise information from the text. This paper makes the following key contributions:

1. **Indic-QA BENCHMARK:** We release a multilingual evaluation benchmark for assessing the Indic Question-Answering capabilities of LLMs, focusing on low-resource languages and multi-domain abstractive tasks.
2. **Empirical Evaluation:** We critically evaluate several esteemed LLMs for Indic languages, comparing their performance on the new benchmark to determine their QA skills.
3. **Translate-Test Paradigm:** We conduct an empirical study using the Translate-Test approach and direct generation in source languages with multilingual LLMs, demonstrating that the Translate-Test approach offers

competitive and often superior performance for low-resource languages.

## 1 Related work

In the realm of context-grounded question answering (QA), significant research has been conducted in both English and Indian languages. This task involves presenting a question along with a contextual paragraph to the model, which then extracts the phrase from the paragraph. Various benchmarks (Dzendzik et al., 2021; Rajpurkar et al., 2016, 2018) have been established for this task, with encoder-only transformer models proving effective in extracting the span containing the answer from the paragraph. The Indic QA community has demonstrated remarkable performance using models like XLM-RoBERTa(Conneau et al., 2019) and others, particularly for multilingual Indian languages. They have a rich dataset to showcase their benchmarks, including SQuAD (Rajpurkar et al., 2016) for English, along with its translated version in Hindi. Additionally, instead of translation, there are datasets specifically designed for evaluating benchmarks in Hindi, such as the Chaii<sup>2</sup> dataset and IndicQA, which are also discussed in this survey paper (Kolhatkar and Verma, 2023). Although there are a few benchmarks for Indic Question Answering, they lack extensive domain coverage, which is crucial for evaluating the robustness of models. In contrast, English benchmarks encompass a wide range of domain-specific datasets such as Resources like the llama Index (Liu, 2022) highlight that selecting the appropriate evaluation dataset is challenging and highly dependent on the specific use case. Academic benchmarks such as BEIR(Thakur et al., 2021) and HotpotQA(Yang et al., 2018) often fail to generalize across different use cases. For example, parameters that work well on certain data domains (e.g., SEC filings) may not perform as effectively on others (e.g., research papers). This challenge led them to the create a dataset hub specifically designed for evaluating RAG systems, encompassing a wide range of domains including research papers, blockchain articles, and code articles.

Additionally, NQ Open (Lee et al., 2019) contains a wealth of Wikipedia content across various domains, and MS MARCO (Bonifacio et al., 2021) features questions sampled from real-world

<sup>2</sup><https://www.kaggle.com/competitions/chaii-hindi-and-tamil-question-answering>

Datasets	As	Bn	Gu	Hi	Kn	MI	Mr	Od	Pa	Ta	Te
Hindi Squad	3099	3107	3371	4734	3068	2926	3165	3079	3469	2743	2955
NQ Open	1462	1483	1570	1842	1447	1420	1511	1451	1570	1331	1420
Chaii	339	351	394	<u>746</u>	351	328	373	305	388	325	361
Indic QA	<u>1789</u>	<u>1763</u>	1369	<u>1547</u>	<u>1517</u>	<u>1589</u>	<u>1604</u>	<u>1680</u>	<u>1542</u>	<u>1804</u>	<u>1734</u>
XSquad	<u>1190</u>	<u>1190</u>	<u>1190</u>	<u>1190</u>	<u>1190</u>	<u>1190</u>	<u>1190</u>	<u>1190</u>	<u>1190</u>	<u>1190</u>	<u>1190</u>
XORQA	537	538	532	537	534	533	529	529	531	537	538
MLQA	2362	2403	2718	<u>4918</u>	2299	2128	2433	2370	2730	2129	2291
Synthetic MCQA*	1741	1662	2162	3802	1618	1248	1807	1753	2326	1150	1416
MS Marco*	29724	30089	31741	35735	29212	28528	30180	30073	32032	27197	28995
Llama Index*	1158	1312	1333	1384	1310	1250	1316	1263	1175	1258	1306

Table 1: Indic-QA Dataset Statistics. Indic-QA benchmark is a compilation of existing datasets, English datasets translated to Indian languages, and, synthetic dataset generated using Gemini. The dataset comprises Extractive Question Answering and abstractive Question Answering (\*). As: Assamese, Bn: Bengali, Gu: Gujarati, Hi: Hindi, Kn: Kannada, MI: Malayalam, Mr: Marathi, Od: Odia, Pa: Punjabi, Ta: Tamil, Te: Telugu. We have translated NQ Open, XORQA, Llama Index, and MS Marco datasets to Hindi. We have translated all the above datasets to the remaining ten Indian languages (underline data instances were already present in the referred language.)

user searches with contexts derived from web documents. The diversity of user queries leads to a broad range of content, making MS MARCO highly versatile. Although initially intended for a different task, we adopted this dataset for our purposes. Hence to address the lack of existing Indic QA benchmark datasets, we translated and adapted several commonly used English QA datasets into 11 Indic languages. This approach provides a more comprehensive and robust evaluation framework for Indic Question Answering models. By leveraging these datasets, we aim to offer a diverse and extensive evaluation resource, enhancing the development and assessment of QA models in Indic languages.

Earlier attempts at the Translate-Test approach (Etzaniz et al., 2023; Intrator et al., 2024) faced limitations due to less advanced translation systems. However, the emergence of larger bilingual parallel datasets (Reid and Artetxe, 2022) has allowed researchers to develop robust neural translation models, greatly improving translation performance. To the best of our knowledge, no one has Tried the Translate-Test approach for Indic QA systems.

## 2 Benchmarks

The primary focus of this work is on context-based QA, where the answer is found within the given context. The datasets utilized in this study were tailored to facilitate this task, with each instance composed of triples consisting of a context, a question, and an answer. This section provides a detailed description of the methodology used to create or modify the existing dataset for our task.

### 2.1 Datasets

In this section, we provide a catalog of the datasets constituting this benchmark, complete with a thorough exposition of their original accessibility and the modifications we have implemented. These datasets are either pre-existing or have been released as part of this work. Following is a detailed description of each dataset.

1. **Hindi SQuAD**: This dataset is a translated version of the original SQuAD (Rajpurkar et al., 2016) into Hindi. It consists of nearly 5,000 instances, translated using the Google Translate API. We translated that from Hindi to other Indic languages.
2. **XQuAD**: XQuAD (Cross-lingual Question Answering Dataset) (Artetxe et al., 2019) is a benchmark for evaluating cross-lingual question answering performance. It consists of 240 paragraphs and 1,190 question-answer pairs sourced from the SQuAD v1.1 development set (Rajpurkar et al., 2016), with professional translations into ten languages. However, we use the version from (Singh et al., 2024), which includes manual translations for all Indic languages.
3. **ChaII Dataset** (Thirumala and Ferracane, 2022): This question-answering dataset features context-question-answer triples in Hindi and Tamil, gathered directly without translation. Created by expert data annotators who are native speakers, the dataset presents a realistic information-seeking task focused on

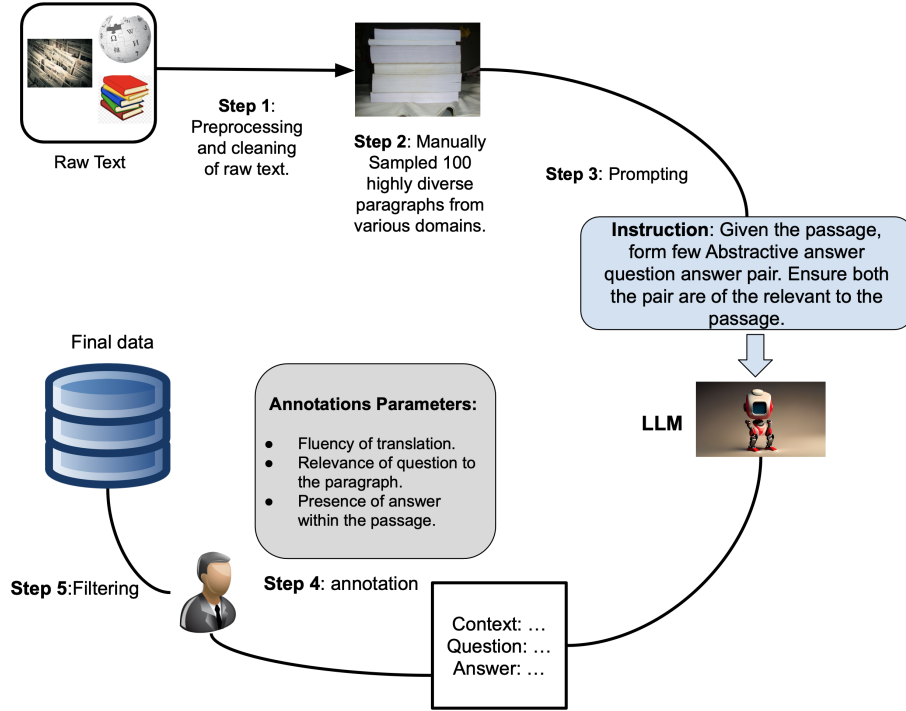


Figure 3: Workflow of synthetic data creation, in the fig. LLM used is Gemini-pro model.

predicting answers to genuine questions about Wikipedia articles. It was used in a Kaggle challenge and includes 1104 questions in Hindi and Tamil, we used the Hindi part of the data and translated it to 10 other Indian languages.

4. **Indic QA** (Doddapaneni et al., 2022): This dataset is a manually curated cloze-style reading comprehension dataset designed for evaluating question-answering models in 10 Indic languages Since this dataset doesn't have Gujarati translation we translated it from Hindi to Gujarati and validated the translation as described in the [2.2] section.
5. **MLQA** (Lewis et al., 2019): MLQA (Multi-Lingual Question Answering) is a benchmark dataset for evaluating cross-lingual question answering performance. We have used the MLQA test set for benchmarking purposes, the test set contains 4918 triples of the form (context, question, answer) all available in Hindi, hence we translated this triplet from Hindi to 10 other Indian languages.
6. **MS Marco** (Bonifacio et al., 2021): Microsoft Machine Reading Comprehension (MS MARCO) is a collection of large-scale datasets designed for deep learning applica-

tions related to search. The questions in MS Marco are sampled from real, anonymized user queries. The context passages, from which the answers are derived, are extracted from real web documents using the latest version of the Bing search engine. We initially considered adapting the multilingual version of the MS MARCO passage ranking dataset (mMarco) for our setting. However, since mMarco lacks a test set, we opted to use the MS MARCO test set, which contains 100k instances, each including a query and a set of passages, among which only one is relevant to the query. We filtered out instances without any relevant passages, resulting in a dataset of 55k instances.

We then translated this dataset from English to Hindi. After applying certain filtering conditions, The exact steps are detailed in [2.2]. The final dataset now includes the question, the source document, and the corresponding answer, and is available in 11 Indian languages.

7. **NQ-Open trans** (Lee et al., 2019): The NQ-Open task is an open-domain question-answering benchmark derived from Natural Questions. The objective is to predict an English answer string for a given English ques-



tion, with all questions answerable using the contents of English Wikipedia. Initially, the dataset was entirely in English, with context, question, and answer all in English. The context often included tables scraped from HTML pages of Wikipedia, resulting in numerous HTML tags. To clean the dataset, we removed all triples where the context contained a table and eliminated all other HTML tags from the remaining examples. In this modified dataset, the fields include the source document (the entire Wikipedia page), the long answer (a paragraph from the page containing the answer), and the exact phrase or word from that paragraph as the short answer. We modified the long answer to serve as the context and the short answer as the answer for the corresponding question. and Since after all this modification dataset was in English we translated that to other Indian languages.

8. **XORQA** (Asai et al., 2020): Cross-lingual Open Retrieval Question Answering (XOR QA) consists of three tasks involving cross-lingual document retrieval from both multilingual and English resources. This dataset was subsequently translated into other Indian languages by (Singh et al., 2024). We utilized the same since it was cross-lingual data, the context was in English while the questions and answers were in other languages. To adapt it to our setting, we translated the context into various Indian languages.
9. **LLama Index**<sup>3</sup> (Liu, 2022): The dataset includes question-answer pairs along with source context, serving as an evaluation tool for the RAG pipeline. We observed that some contexts were insufficient to answer the questions effectively. To address this, we applied the BGE-M3 (Chen et al., 2024) algorithm to measure the similarity between the context and the query, using a threshold of 0.43 to determine if a question could be answered adequately based on the context. Post filtering we translated the resulting context, question, and answer triplets into Hindi and Hindi other Indian languages.
10. **Synthetic Data**: This dataset is introduced as part of this study. We employed the Gem-

ini model (Team et al., 2023) to generate question-answer pairs based on provided contexts. To achieve this, we sampled a diverse set of Hindi contexts from sources such as Wikipedia, storybooks, Indian news articles, and paragraphs from competitive exams. We then prompted the model with these context paragraphs to generate abstractive question-answer pairs, framing the task as an abstractive QA task. Subsequently, this dataset was translated into other languages and verified by language experts, the whole workflow process can be found [3] [A.1].

## 2.2 Data Curation Methodology

In light of the approaches discussed previously in Section 1, context-grounded question-answering datasets can generally be categorized into two types: abstractive and extractive. Although many extractive datasets exist for high-resource languages, the few available for Indian languages lack diversity in domains and question types, limiting their usefulness for benchmarking. Hence, we extended the benchmark suite available in English to these Indian languages by translating. We utilized IndicTrans2<sup>4</sup> (Gala et al., 2023) for translation, an open-source transformer-based multilingual NMT model that supports high-quality translations across all the 22 scheduled Indian languages. We segmented the context paragraph into sentences using the Spacy library, translated each sentence, and then recombined them. This approach yielded better translation results, and importantly, the model did not lose context when translating, thus preserving the coherence of the text. In the list of datasets for benchmarking, some are available only in English (e.g., NQ-open, ORQA, llama index, MS-Marco), while others are available in both English and Hindi (e.g., Hindi SQuAD, CHAI, MLQA, Synthetic data). Additionally, a few datasets (e.g., IndicQA, XSQuAD) are also available in all 10 or 11 languages with verified translations. For all the datasets not found in the respective language, we translated them and applied the filtering methods discussed below.

To assess the quality of our translations, we initially translated each dataset from the source language to the target language, followed by back-translation from the target language back to the source language. We then calculated the CHRF

<sup>3</sup><https://www.llamaindex.ai/blog/introducing-llama-datasets-aadb9994ad9e>

<sup>4</sup><https://github.com/AI4Bharat/IndicTrans2>

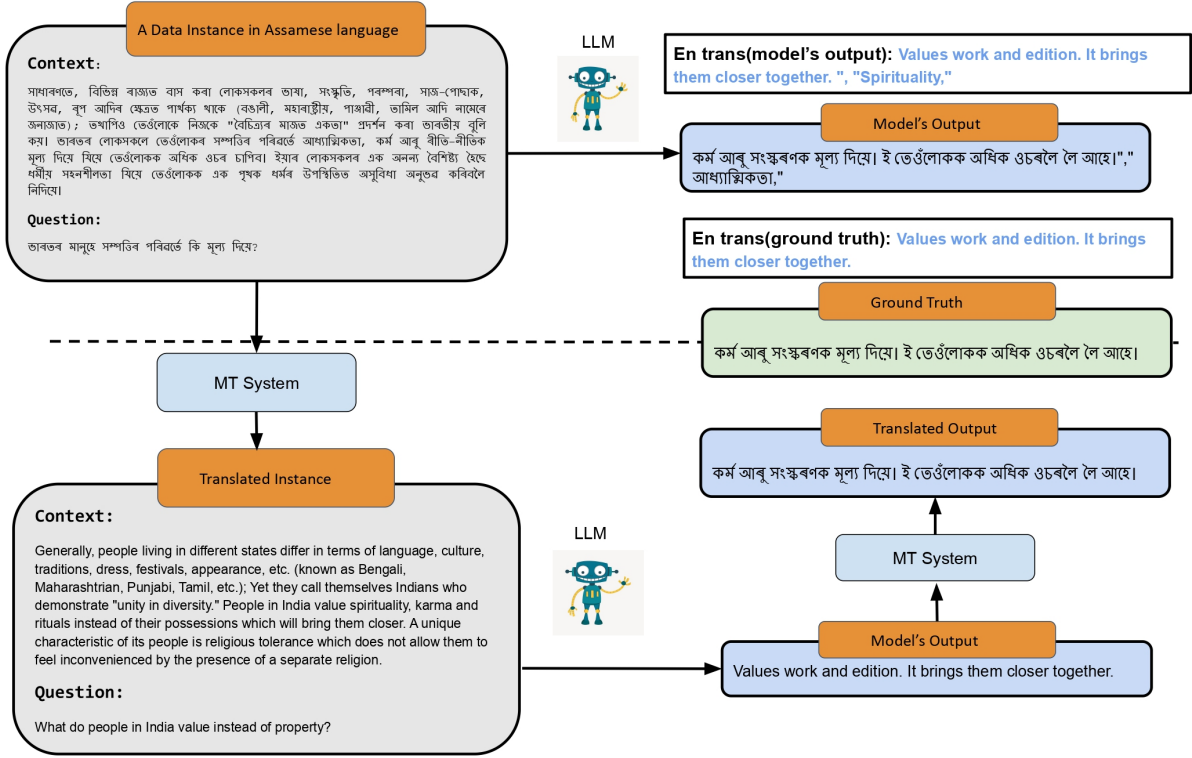


Figure 4: Inference in source Language (Top) vs Translate Test Inference (Bottom).

scores(Popović, 2015) between the original and back-translated sentences, using a threshold of 50 to filter the instances. Additionally, we manually verified a subset of the filtered data to ensure accuracy. For the translation process, we initially translated the English data directly into Hindi. After filtering the data, we then translated it from Hindi to other Indian languages, rather than directly from English. This approach was based on our observation that the translation quality from Hindi to other Indian languages was superior. The improved quality can be attributed to the linguistic similarities within the same language family, including morphology, syntax, and grammar.

### 3 Experiments

We conducted a series of experiments to evaluate the performance of existing LLMs, using NVIDIA A100 GPUs in both 40GB and 80GB variants for our computational needs. Our computational needs signify GPUs both for Translation and evaluation over the models. For inference we utilized VLLM (Kwon et al., 2023) which is an open-source library that supports LLM inference efficiently.

We evaluate the following LLMs on our benchmark: OpenHathi<sup>5</sup> and its instruction-finetuned

variant (IFV) known as Airavata (Gala et al., 2024), Bloom (Le Scao et al., 2022) and its IFV named Bloomz, Gemma(Team et al., 2024), and its instruction fine-tuned variant Gemma-IT. OpenHathi (7B parameter model), was created through continual pre-training on the LLaMA-2 model (Touvron et al., 2023). Airavata (Gala et al., 2024) (7B parameter model) is an instruction fine-tuned version of OpenHathi. Both OpenHathi and Airavata are specifically trained for Hindi.

Gemma and Gemma-IT (7B parameter models)<sup>6</sup> were released by Google. While these models are not specifically trained for Indian languages, they demonstrate multilingual capabilities. Additionally, Aya-8B (Aryabumi et al., 2024) is an instruction-tuned model specifically designed for multilingual applications. We also tested Aya-101, another instruction-tuned model based on MT5, which is available in a 13B size. Furthermore, we explored the LLaMA {3, 3.1} and LLaMA {3, 3.1} Instruct models<sup>7</sup>, which are 8B parameter models and part of the LLaMA family. LLaMA-3 has been trained on data from approximately 30 languages, excluding English. we also used Narvasa2.0<sup>8</sup> and

OpenHathi-7B-Hi-v0.1-Base

<sup>6</sup><https://ai.google.dev/gemma/docs>

<sup>7</sup><https://ai.meta.com/blog/meta-llama-3/>

<sup>8</sup>[huggingface.co/Telugu-LLM-Labs/](https://huggingface.co/Telugu-LLM-Labs/)

<sup>5</sup><https://huggingface.co/sarvamai/>

Languages	Bloom		Gemma		Llama-3		Llama3.1		Gemma-2	
	Direct Inference	Translate	Direct Inference	Translate	Direct Inference	Translate	Direct Inference	Translate	Direct Inference	Translate
<b>As</b>	13.86	<b>16.19</b>	21.70	16.08	19.23	<b>24.14</b>	21.29	<b>22.48</b>	33.01	<b>35.60</b>
<b>Bn</b>	17.84	16.07	21.15	16.69	19.14	<b>23.96</b>	27.61	24.00	38.33	35.11
<b>Gu</b>	13.27	<b>18.59</b>	24.90	<b>25.12</b>	20.27	<b>29.55</b>	24.08	<b>32.11</b>	40.10	<b>44.86</b>
<b>Hi</b>	21.69	19.29	34.37	19.72	41.53	30.02	46.76	26.22	44.46	43.18
<b>Kn</b>	15.63	<b>17.86</b>	24.47	16.31	20.39	<b>25.84</b>	21.31	<b>26.22</b>	35.31	<b>37.98</b>
<b>MI</b>	19.22	17.96	25.20	17.31	22.80	<b>28.07</b>	31.61	29.49	38.35	<b>41.54</b>
<b>Mr</b>	15.12	<b>18.89</b>	23.96	17.46	36.12	27.54	39.98	31.46	41.89	41.10
<b>Od</b>	11.11	<b>15.09</b>	9.06	<b>15.52</b>	14.20	<b>23.76</b>	12.78	<b>24.37</b>	28.81	<b>36.39</b>
<b>Pa</b>	15.60	<b>20.54</b>	28.43	19.88	21.96	<b>30.54</b>	32.04	29.03	43.37	<b>45.10</b>
<b>Ta</b>	19.96	18.22	22.45	17.42	19.74	<b>27.54</b>	28.04	<b>28.06</b>	39.64	<b>40.29</b>
<b>Te</b>	16.07	<b>17.79</b>	23.93	17.72	13.59	<b>25.37</b>	22.13	<b>27.04</b>	34.97	<b>39.24</b>

Table 2: Performance of both Direct Inference and Translate-Test Inference for various Large Language Models on the Zero-Shot Extractive Indic QA Benchmark. We report the average F1 scores across span-extraction datasets. Instances where Translate-Test outperforms Direct Inference are indicated in **bold**.

Languages	Bloom		Openhathi		Llama-3		Llama-3.1		Gemma		Gemma-2	
	Direct Inference	Translate	Direct Inference	Translate	Direct Inference	Translate	Direct Inference	Translate	Direct Inference	Translate	Direct Inference	Translate
<b>As</b>	5.91	<b>11.61</b>	0.33	<b>3.98</b>	3.97	<b>7.69</b>	4.17	<b>10.13</b>	6.98	<b>7.78</b>	11.32	<b>14.23</b>
<b>Bn</b>	8.21	<b>11.35</b>	0.47	<b>3.58</b>	4.40	<b>7.71</b>	5.10	<b>9.93</b>	6.48	<b>7.22</b>	10.88	<b>13.85</b>
<b>Gu</b>	8.73	<b>12.00</b>	0.12	<b>4.20</b>	6.97	<b>7.98</b>	6.05	<b>10.25</b>	9.06	7.14	12.78	<b>14.64</b>
<b>Hi</b>	8.44	<b>12.14</b>	5.19	4.04	11.17	8.21	11.48	10.66	8.63	7.90	12.33	<b>14.84</b>
<b>Kn</b>	8.19	<b>11.87</b>	0.15	<b>3.76</b>	4.88	<b>7.70</b>	4.71	<b>10.19</b>	7.53	<b>8.04</b>	12.34	<b>14.78</b>
<b>MI</b>	8.39	<b>11.68</b>	1.01	<b>4.13</b>	6.75	<b>7.66</b>	7.99	<b>9.94</b>	7.14	<b>7.56</b>	13.39	<b>14.27</b>
<b>Mr</b>	8.39	<b>11.38</b>	1.71	<b>3.85</b>	10.08	7.79	9.45	<b>9.93</b>	8.23	7.49	11.98	<b>14.23</b>
<b>Od</b>	6.84	<b>11.58</b>	0.00	<b>3.68</b>	7.01	<b>7.61</b>	7.88	<b>9.93</b>	3.38	<b>6.77</b>	11.70	<b>14.12</b>
<b>Pa</b>	7.91	<b>11.90</b>	0.15	<b>4.18</b>	6.10	<b>8.09</b>	5.81	<b>10.28</b>	9.72	7.84	12.96	<b>14.90</b>
<b>Ta</b>	9.52	<b>12.66</b>	0.47	<b>4.34</b>	3.72	<b>7.95</b>	6.17	<b>10.46</b>	7.74	<b>7.94</b>	13.67	<b>15.38</b>
<b>Te</b>	8.96	<b>12.40</b>	0.19	<b>3.90</b>	3.04	<b>8.10</b>	5.56	<b>10.64</b>	9.34	8.14	13.80	<b>15.19</b>

Table 3: Performance of both Direct Inference and translate test inference of various Large Language Models on Zero-Shot Abstractive INDIC QA BENCHMARK. We report the average numbers (Rouge-L) scores across the languages. Instances where Translate-Test outperforms Direct Inference are indicated in **bold**.

Mistral NeMo<sup>9</sup> because of their strong multilingual performance. This diverse range of models enables a thorough evaluation of the strengths and performance of our benchmarks across different architectures and training methods.

### 3.1 Evaluation Metrics

We chose widely used QA evaluation metrics for evaluating both extractive and abstractive Question Answering datasets:

**1. F1 (macro-averaged) score:** This score represents the harmonic mean of precision and recall, calculating the average similarity between predicted and actual answers by comparing the sets of words or tokens in the predicted and ground truth sentences.

**2. ROUGE(L):** A Recall-Oriented Understudy to measure how much of the words from the gold response are present in the generated response. This

metric is commonly used for generative tasks such as summarization, and we have used it to evaluate abstractive QA tasks.

### 3.2 Translate-Test Inferences

In addition to direct inference, we conducted experiments on translate-test<sup>4</sup> inferences involving the following steps:

1. Use **IndicTrans2** to translate the source language input (context and question) to English.
2. Prompt the LLM with the translated input and get the response.
3. Back-translate the generated LLM’s English response to the source language.

This method ensures the evaluation of the LLMs’ performance across languages by utilizing translation systems to bridge the gap between languages during inference.

Indic-gemma-7b-finetuned-sft-Navarasa-2.0

<sup>9</sup><https://mistral.ai/news/mistral-nemo/>

Languages	Bloom		Gemma		Llama-3		Openhathi	
	1 shot	3 shot	1 shot	3 shot	1 shot	3 shot	1 shot	3 shot
<b>As</b>	21.22	22.44	37.91	40.43	28.61	29.42	3.15	3.14
<b>Bn</b>	19.82	26.90	42.64	37.85	32.71	24.95	6.86	6.56
<b>Gu</b>	20.17	22.16	44.26	47.40	26.76	22.46	3.65	4.61
<b>Hi</b>	34.08	36.96	54.95	56.95	54.79	57.11	10.27	22.1
<b>Kn</b>	19.99	19.93	37.74	35.80	22.76	15.60	0.33	2.26
<b>MI</b>	25.82	23.84	42.08	31.10	28.94	22.60	6.59	5.50
<b>Mr</b>	26.35	25.99	47.08	37.39	47.17	40.7	5.82	8.63
<b>Od</b>	15.72	15.68	16.66	22.05	13.37	5.39	0.37	3.32
<b>Pa</b>	23.18	23.98	42.18	47.17	31.03	30.62	2.00	3.52
<b>Ta</b>	26.80	26.97	43.27	43.90	28.66	27.06	4.51	4.94
<b>Te</b>	20.83	25.22	39.90	43.62	25.37	20.86	3.09	2.76

Table 4: Performance of various Large Language Models on few-Shot Extractive INDIC QA BENCHMARK. We report the average (F1-score) across span-extraction datasets and question-answering datasets.

## 4 Result and analysis

**Comparing Base Model Performance and Effect of Few Shot:** Table 6 shows the base LLMs’ performance in the zero-shot setting. The Gemma model excels in extractive question-answering tasks, surpassing the Bloom and Llama-3 base models. However, Llama-3 outperforms Gemma in Hindi, Marathi, and Odia languages. Bloom surpasses all models in abstractive question-answering tasks and for all languages. Notably, base models generally perform poorly on abstractive question-answering datasets compared to extractive ones.

We also evaluate the effect of in-context examples as reported in Table 4. As expected, using few-shot (1-shot and 3-shot) almost always improves over the zero-shot base model. However, we can spot some language-specific patterns where Bloom and Openhathi behave differently than Gemma and Llama-3. For example, for some languages such as *Bn*, *MI*, *Mr* Gemma and Llama-3 show a significant drop with the increase in few shot examples, however, Bloom and Openhathi retain or even improve. We believe this is correlated with the availability of language-specific corpus and their utilization in training these models.

**Effect of Instruction Finetuning:** Table 6 shows the performance of instruction-finetuned models in our study. Instruction finetuning generally improves abstractive QA tasks across all models, but its impact on extractive QA varies. For instance, Gemma and Llama-3 perform better than Bloom and OpenHathi in their base models, but their instruction-finetuned variants do not show sig-

nificant improvement. This is because these models were primarily instruction-finetuned on non-Indic languages, which compromises their generic multilingual ability during task-specific finetuning, leading to lower results.

On the other hand, OpenHathi was specifically trained on the Hindi language and so is its instruction finetuning variant Airavata. As a result, the performance of OpenHathi is significantly poor in all languages. Airavata benefits from further instruction finetuning on Hindi data and improves over OpenHathi for Hindi language but suffers poorly for other Indian languages. Bloomz produces the highest jump compared to Bloom and we hypothesize this is because a good portion of evaluation benchmark coming from generic-domain such as Wikipedia data has been seen by Bloomz during its training and instruction finetuning, making it a good choice for applications which aims to use common world knowledge.

**Extractive vs Abstractive Tasks:** While it is clear that instruction finetuning helps more in abstractive QA tasks, both Table 5 and Table 6 show a positive correlation between the scores for extractive tasks and abstractive tasks across languages. This is almost true for all the base and instruct variants of the models except Gemma, where Gemma instruct improves the abstractive QA score but deteriorates in the extractive QA task. Careful analysis shows that abstractive task metrics change moderately between base models and their instruction finetuned variants. This is expected because abstractive metrics such as Rouge-L are



more heuristic-driven in nature and designed to ignore small variations, natural in non-deterministic text generation, unlike extractive metrics such as F1. Thus abstractive metrics deviate on a smaller scale than extractive metrics. However, the positive correlation between both task metrics across models clearly establishes that the factors affecting the overall performance of the models show similar signs for both extractive and abstractive tasks and hence improving one will likely improve the other as well.

**On How to Choose a Model:** Going by the results so far, one would pick BloomZ if the application needs only common world knowledge and needs a model which does well out-of-the-box. If there is a use-case for which we have adequate Indic language finetuning data, it might be good to build over the world knowledge acquired by Gemma and Llama-3 and do instruction finetuning on Indic languages to make it better suitable for abstractive QA tasks. If we are very specific about a certain niche domain in only the Hindi language, where common world knowledge is not a pre-requisite, Airavata can be a good candidate given its focus on Hindi-based training and improvements in both extractive and abstractive tasks with instruction finetuning. However, the Aya models are particularly well-suited if we are specifically seeking high-quality instruction-tuned models for Indian languages.

**Translate Test is an Effective Alternative to Source Language:** Translation-based approaches are often more effective than direct generation in source languages using multilingual models. For languages like Punjabi, Gujarati, and Oriya, the translation-based approach outperforms the multilingual model. This comparison divides languages into two categories: (1) those where multilingual models perform better and (2) those where translation-based approaches are superior.

Mid-resource languages like Hindi and Bengali benefit more from multilingual models, while low-resource languages like Oriya and Punjabi perform better with translation-based approaches. This is because multilingual models struggle with insufficient language-specific data, leading to poor performance. In contrast, translation-based approaches leverage high-resource languages for reasoning and generation, making it easier to learn the translation task from multilingual data.

## 5 Conclusion

In this paper, we present a benchmark for evaluating the grounded Question-Answering (QA) capabilities of Large Language Models (LLMs) on both extractive and abstractive tasks. Our findings reveal that instruction-tuning with target language data significantly enhances QA performance, while the Translate-Test technique yields better results for low-resource languages. In contrast, high-resource languages benefit more from source language inference due to larger training datasets.

Despite advances in multilingual training, LLMs struggle with transfer learning to low-resource languages. Integrating effective translation systems into the QA pipeline is crucial for improving performance in these contexts. By releasing this benchmark, we aim to promote further research into the QA capabilities of LLMs across various languages, particularly for those that are underrepresented.

## 6 Limitation

Our research aims to provide a challenging and comprehensive benchmark for evaluating LLMs on the Hindi QA task, but it does face several limitations.

(1) The availability of high-quality datasets for Hindi is limited. Despite our best efforts to curate the benchmark from various sources, there might still be an inherent bias introduced during the data collection and translation process. Additionally, although we conducted quality checks, there may be subjective interpretability issues with the translated datasets. (2) while we attempted to diversify across various domains, the benchmark may not depict the true performance in a completely unseen domain.

Despite the strong performance of the Translate Test technique, particularly for low-resource languages, it has several limitations. One significant drawback is the potential for cascading errors. Translation errors occurring early in the pipeline can propagate through subsequent stages, adversely affecting the final output. This issue is typical in models that rely on sequential processing, where initial inaccuracies can compound over time. Moreover, while the Translate Test approach currently shows promising results, the ideal solution is the development of robust multilingual models that can handle and mitigate cascading errors effectively. Such models should be capable of generalizing well across various languages without relying heavily on error-prone translation processes. Our analysis fo-

cused on base models rather than instruction-tuned models. Previous research has shown that multilingual instruction-tuned A well-optimized multilingual instruction-tuned model could potentially address these limitations.

However, the challenge remains in developing effective instruction-tuning data for low-resource languages, whether through translation or other methods. This underscores the need for continued research to enhance instruction-tuning strategies in multilingual settings.

## Acknowledgments

We thank the anonymous reviewers for their constructive feedback. This work is supported by and is part of BharatGen (<https://bharatgen.tech/>), an Indian Government-funded initiative focused on developing multimodal large language models for Indian languages.

## References

- Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. *arXiv preprint arXiv:1910.11856*.
- Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Jon Ander Campos, Yi Chern Tan, Kelly Marchisio, Max Bartolo, Sebastian Ruder, Acyr Locatelli, Julia Kreutzer, Nick Frosst, Aidan Gomez, Phil Blunsom, Marzieh Fadaee, Ahmet Üstün, and Sara Hooker. 2024. [Aya 23: Open weight releases to further multilingual progress](#). *Preprint*, arXiv:2405.15032.
- Akari Asai, Jungo Kasai, Jonathan H Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. 2020. Xorqa: Cross-lingual open-retrieval question answering. *arXiv preprint arXiv:2010.11856*.
- Luiz Bonifacio, Israel Campiotti, Roberto de Alencar Lotufo, and Rodrigo Frassetto Nogueira. 2021. [mmarco: A multilingual version of MS MARCO passage ranking dataset](#). *CoRR*, abs/2108.13897.
- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. [Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#). *Preprint*, arXiv:2402.03216.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.
- Sumanth Doddapaneni, Rahul Aralikkatte, Gowtham Ramesh, Shreyansh Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2022. Towards leaving no indic language behind: Building monolingual corpora, benchmark and models for indic languages. *ArXiv*, abs/2212.05409.
- Daria Dzendzik, Carl Vogel, and Jennifer Foster. 2021. English machine reading comprehension datasets: A survey. *arXiv preprint arXiv:2101.10421*.
- Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lopez de Lacalle, and Mikel Artetxe. 2023. Do multilingual language models think better in english? *arXiv preprint arXiv:2308.01223*.
- Jay Gala, Pranjal A Chitale, Raghavan AK, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar, Janki Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, et al. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *arXiv preprint arXiv:2305.16307*.
- Jay Gala, Thanmay Jayakumar, Jaavid Aktar Husain, Aswanth Kumar M, Mohammed Safi Ur Rahman Khan, Diptesh Kanojia, Ratish Puduppully, Mitesh M. Khapra, Raj Dabre, Rudra Murthy, and Anoop Kunchukuttan. 2024. Airavata: Introducing hindi instruction-tuned llm. *arXiv preprint arXiv:2401.15006*.
- Yotam Intrator, Matan Halfon, Roman Goldenberg, Reut Tsarfaty, Matan Eyal, Ehud Rivlin, Yossi Matias, and Natalia Aizenberg. 2024. Breaking the language barrier: Can direct inference outperform pre-translation in multilingual llm applications? *arXiv preprint arXiv:2403.04792*.
- Dhruv Kolhatkar and Devika Verma. 2023. Indic language question answering: A survey. In *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, pages 697–703. IEEE.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.
- Teven Le Scao, Angela Fan, Christopher Akiki, Elie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model.
- Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. [Latent retrieval for weakly supervised open domain question answering](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

- Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.
- Jerry Liu. 2022. [LlamaIndex](#).
- Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. [The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only](#). *Preprint*, arXiv:2306.01116.
- 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.
- Mohammed Safi Ur Rahman Khan, Priyam Mehta, Ananth Sankar, Umashankar Kumaravelan, Sumanth Doddapaneni, Varun Balan G, Sparsh Jain, Anoop Kunchukuttan, Pratyush Kumar, Raj Dabre, et al. 2024. IndicLLMsuite: A blueprint for creating pre-training and fine-tuning datasets for indian languages. *arXiv e-prints*, pages arXiv–2403.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don’t know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.
- Machel Reid and Mikel Artetxe. 2022. On the role of parallel data in cross-lingual transfer learning. *arXiv preprint arXiv:2212.10173*.
- Harman Singh, Nitish Gupta, Shikhar Bharadwaj, Dinesh Tewari, and Partha Talukdar. 2024. Indicgenbench: A multilingual benchmark to evaluate generation capabilities of llms on indic languages. *arXiv preprint arXiv:2404.16816*.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. [BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models](#). In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Adhitya Thirumala and Elisa Ferracane. 2022. Extractive question answering on queries in hindi and tamil. *arXiv preprint arXiv:2210.06356*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *Preprint*, arXiv:2307.09288.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

## A Appendix

### A.1 Synthetic Data Creation

As discussed in 2.1, in addition to the translation dataset from high-resource to low-resource languages, we also generated data using the Gemini model (Team et al., 2023). We chose this model due to its strong multilingual performance and considerable parameter count. To start, we sampled paragraphs from various sources, including Wikipedia, storybooks, Indian news articles, and a selection of books, aiming for domain diversity while incorporating Indian cultural nuances. We developed our benchmark in Hindi and selected a subset of the generated data for human verification. The verification process focused on several parameters, such as the fluency of the generated questions and answers, and whether the questions were directly derived from the paragraphs. Given that large language models can hallucinate even with clear instructions annotators were asked to score each instance on a scale of 1 to 5, where 1 indicated irrelevance

and 5 indicated exact relevance. We also evaluated whether the answers were generated from the text or relied on external knowledge.

Once verified, we translated these data instances into other Indian languages. The synthetic data primarily consisted of texts from Ramayana textbooks, recent Indian news articles, cultural articles from Wikipedia, and classic storybooks, Dharmapal books. Most of the data was extracted from books using Optical Character Recognition (OCR), followed by a thorough cleaning process to compile useful passages.

## A.2 Analysis

### **Indic Languages are Medium or Low Resource:**

Before diving into the results, it's important to consider the resource availability in Indic languages. This influences the best strategy for QA tasks in these languages. Figure 1 shows the approximate number of tokens available for each language from sources like Wikipedia, websites, and PDFs. Compared to English, which has trillions of tokens, Indic languages have far fewer resources. Languages like Hindi and Bengali are medium-resource, while most others are low-resource, with some like Oriya and Punjabi being very low-resource. Although the statistics in Figure 1 are estimates and change over time and across different models, classifying languages into high, medium, and low-resource groups is important for our analysis.

Tables 2 and ?? present important observations, which we outline below:

**Extractive vs Abstractive Shows Same Pattern:** The patterns shown by both the extractive and abstractive dataset results in Tables 2 and ?? are consistent. This is due to the impact of the availability or lack of language-specific training data, and the quality of translation remaining the same for both types of QA tasks. Therefore, the insights and patterns observed in Table 2 also apply to Table ??.

**Gemma2-Base performed best on our experiments:** There is noticeable variation in the performance of different LLMs across languages. This can be attributed to the varying amounts of language-specific training data and the different sizes of the models. Despite this variation, the observation that translation-based methods work better for low-resource languages generally holds true across all LLMs.

Among the LLMs, Gemma2 is the clear winner across most languages and tasks. Llama-3 and

Gemma perform similarly and slightly better than Bloom. The performance ranking of LLMs tends to follow their release order, suggesting that more training data improves their multilingual capabilities. Openhathi performs relatively poorly compared to other models, likely because it has only been trained on Hindi data and has limited exposure to other languages.

Additionally, Aya-101, Narvosa 2.0, and Mistral NeMo are superior instruction-tuned models. This can be attributed to their fine-tuning on Indic data, whereas other models perform well in their base variants but struggle during instruction fine-tuning. This difficulty is likely due to catastrophic forgetting, as the fine-tuning data for these models contains significantly less Indic data compared to their pre-training datasets. Aya-101 is fine-tuned over the base model mT5, which is an encoder-decoder model (non-LLM). Although it is 13B parameter model which is greater than other in terms of parameter count, then to its performance is way higher than other multilingual instruction tuned models [5].

## Frequently Asked Questions

### **1) Why choose IndicTrans2 over other available translation models?**

Answer: (Gala et al., 2023) show that IndicTrans2 surpasses other models, such as NLLB and Google Translate, particularly for English to Hindi translation tasks. In our analysis, we also tested NLLB, but the CHRF scores for back-translated and source texts were lower than those achieved with the IndicTrans2 model.



Languages	Llam3.1 Instruct		Gemma2-it		Navarasa-2.0		Mistral-nemo		Aya-101	
	Ext	Gen	Ext	Gen	Ext	Gen	Ext	Gen	Ext	Gen
<b>As</b>	13.82	3.33	3.43	1.32	21.59	3.27	21.29	0.93	43.38	3.15
<b>Bn</b>	15.96	3.69	5.33	1.98	19.14	3.59	25.35	1.57	57.64	3.79
<b>Gu</b>	12.73	5.80	6.19	3.37	30.77	6.01	24.08	2.61	61.42	5.84
<b>Hi</b>	25.31	3.93	10.63	2.64	31.31	5.62	46.76	2.03	70.45	2.43
<b>Kn</b>	6.37	4.46	3.35	1.85	24.55	4.99	21.31	1.24	55.21	4.23
<b>MI</b>	14.56	5.88	4.42	2.10	28.40	5.21	31.61	2.13	55.52	5.25
<b>Mr</b>	24.37	4.68	8.48	2.65	22.90	2.78	39.98	1.28	58.98	3.34
<b>Od</b>	14.56	5.04	2.99	0.95	20.75	3.12	0.26	0.60	48.82	3.23
<b>Pa</b>	10.25	4.21	3.59	1.73	25.84	5.88	32.04	2.42	64.19	6.01
<b>Ta</b>	13.95	6.23	5.75	2.97	18.67	6.57	28.04	2.40	58.27	7.07
<b>Te</b>	10.33	5.32	4.09	2.19	21.94	5.89	22.13	1.66	56.60	7.29

Table 5: Performance of various Large Language Models on Zero-Shot on **Subset** of INDIC QA BENCHMARK (MLQA, NQ open, Synthetic data). We report the average numbers (F1) across span-extraction datasets and (Rouge-L)abstractive question-answering datasets.

Languages	Bloomz		Gemma-Instruct		Llama-3-Instruct		Airavata		Aya-8-Instruct	
	Ext	Gen	Ext	Gen	Ext	Gen	Ext	Gen	Ext	Gen
<b>As</b>	<b>38.69</b>	<b>7.36</b>	11.32	9.97	15.22	3.42	3.70	3.41	19.39	7.19
<b>Bn</b>	<b>44.75</b>	<b>8.27</b>	13.03	11.66	17.89	3.90	6.58	4.38	29.79	<b>8.27</b>
<b>Gu</b>	<b>49.06</b>	<b>9.24</b>	7.61	7.74	12.50	2.90	5.15	2.87	26.48	5.82
<b>Hi</b>	<b>62.88</b>	<b>9.06</b>	18.54	7.61	14.95	7.21	44.35	7.21	55.17	8.02
<b>Kn</b>	<b>40.10</b>	8.27	11.94	8.80	15.88	3.13	1.39	1.07	16.22	<b>8.62</b>
<b>MI</b>	<b>43.84</b>	<b>8.55</b>	9.33	5.28	13.67	2.32	7.55	6.80	30.35	<b>8.55</b>
<b>Mr</b>	<b>50.03</b>	8.09	14.15	10.95	13.84	1.87	14.15	5.91	35.83	<b>9.78</b>
<b>Od</b>	<b>37.10</b>	<b>8.78</b>	1.61	0.72	5.59	1.74	1.78	1.27	15.77	5.79
<b>Pa</b>	<b>53.37</b>	9.11	10.78	8.23	18.07	4.72	3.53	2.87	11.81	<b>9.79</b>
<b>Ta</b>	<b>46.43</b>	<b>10.33</b>	16.31	10.42	17.98	3.90	7.66	5.01	36.46	10.26
<b>Te</b>	<b>44.42</b>	<b>8.98</b>	10.85	7.53	15.04	4.13	4.43	3.48	18.82	7.61
<b>Average</b>	46.42	8.73	11.41	8.08	14.60	3.57	9.12	4.03	26.92	8.15

Table 6: Performance of various Large Language Models on Zero-Shot INDIC QA BENCHMARK. We report the average numbers (F1) across span-extraction datasets and (Rouge-L)abstractive question-answering datasets.

Languages	Bloom		Gemma		Llama-3		Openhathi		Gemma-2	
	Direct Inference	Translate	Direct Inference	Translate	Direct Inference	Translate	Direct Inference	Translate	Direct Inference	Translate
<b>As</b>	13.86	<b>16.19</b>	21.70	16.08	19.23	<b>24.14</b>	1.14	<b>5.38</b>	33.01	<b>35.60</b>
<b>Bn</b>	17.84	16.07	21.15	16.69	19.14	<b>23.96</b>	1.49	<b>5.46</b>	38.33	35.11
<b>Gu</b>	13.27	<b>18.59</b>	24.90	<b>25.12</b>	20.27	<b>29.55</b>	4.07	<b>7.15</b>	40.10	<b>44.86</b>
<b>Hi</b>	21.69	19.29	34.37	19.72	41.53	30.02	11.83	8.28	44.46	43.18
<b>Kn</b>	15.63	<b>17.86</b>	24.47	16.31	20.39	<b>25.84</b>	0.31	<b>6.24</b>	35.31	<b>37.98</b>
<b>MI</b>	19.22	17.96	25.20	17.31	22.80	<b>28.07</b>	0.68	<b>6.74</b>	38.35	<b>41.54</b>
<b>Mr</b>	15.12	<b>18.89</b>	23.96	17.46	36.12	27.54	2.07	<b>6.74</b>	41.89	41.10
<b>Od</b>	11.11	<b>15.09</b>	9.06	<b>15.52</b>	11.23	<b>23.76</b>	0.26	<b>5.61</b>	28.81	<b>36.39</b>
<b>Pa</b>	15.60	<b>20.54</b>	28.43	19.88	21.96	<b>30.54</b>	1.31	<b>8.23</b>	43.37	<b>45.10</b>
<b>Ta</b>	19.96	18.22	22.45	17.42	19.74	<b>27.54</b>	0.70	<b>7.03</b>	39.64	<b>40.29</b>
<b>Te</b>	16.07	<b>17.79</b>	23.93	17.72	13.59	<b>25.37</b>	0.53	<b>6.38</b>	34.97	<b>39.24</b>
<b>En</b>	27.41		27.13		34.69		8.55		52.17	

Table 7: Performance of both Direct Inference and translate test inference of various Large Language Models on Zero-Shot Extractive INDIC QA BENCHMARK. We report the average numbers (F1) across span-extraction datasets.

Instruction for Base model

**Answer the following question based on the information in the given passage.:**

**Passage:**आमतौर पर विभिन्न राज्यों में रहने वाले लोग अपनी भाषा, संस्कृति, परंपरा, परिधान, उत्सव, रूप आदि में अलग होते हैं (बंगाली, महाराष्ट्रीयन, पंजाबी, तमिलीयन, आदि के रूप में जाने जाते हैं); फिर भी वो अपने आपको भारतीय कहते हैं जो "विविधता में एकता" को प्रदर्शित करता है।भारत में लोग अपनी संपत्ति के बजाय आध्यात्मिकता, कर्म और संस्कार को महत्व देते हैं जो उन्हें और पास लाता है। अपने अनोखे गुण के रूप में यहाँ के लोगों में धार्मिक सहिष्णुता है जो उन्हें अलग धर्म की उपस्थिति में कठिनाई महसूस नहीं करने देती।

**Question:**भारत में लोगों में विभिन्नता के बावजूद एकता कैसे प्रदर्शित होती है?

**Answer:**

Figure 5: Evaluation prompt used for the base model.

Instruction for Instruction tuned  
model(Airavata)

<s><|system|>

Answer the following question based on the information in the given passage.:

<|user|>

**Passage:** आमतौर पर विभिन्न राज्यों में रहने वाले लोग अपनी भाषा, संस्कृति, परंपरा, परिधान, उत्सव, रूप आदि में अलग होते हैं (बंगाली, महाराष्ट्रीयन, पंजाबी, तमिलीयन, आदि के रूप में जाने जाते हैं); फिर भी वो अपने आपको भारतीय कहते हैं जो "विविधता में एकता" को प्रदर्शित करता है। भारत में लोग अपनी संपत्ति के बजाय आध्यात्मिकता, कर्म और संस्कार को महत्व देते हैं जो उन्हें और पास लाता है। अपने अनोखे गुण के रूप में यहाँ के लोगों में धार्मिक सहिष्णुता है जो उन्हें अलग धर्म की उपस्थिति में कठिनाई महसूस नहीं करने देती।

**Question:** भारत में लोगों में विभिन्नता के बावजूद एकता कैसे प्रदर्शित होती है?

**Answer:**

<|assistant|>

Figure 6: Evaluation prompt used for the instruction-tuned model. Most models utilize a chat format. This is an example of the prompt used for the Airavata model.

Output from Bloom model

Answer the following question based on the information in the given passage.:

**Passage:** विषाणु अकोशिकीय अतिसूक्ष्म जीव हैं जो केवल जीवित कोशिका में ही वंश वृद्धि कर सकते हैं। [1] ये नाभिकीय अम्ल और प्रोटीन से मिलकर गठित होते हैं, शरीर के बाहर तो ये मृत-समान होते हैं परंतु शरीर के अंदर जीवित हो जाते हैं। इन्हे क्रिस्टल के रूप में इकट्ठा किया जा सकता है। एक विषाणु बिना किसी सजीव माध्यम के पुनरुत्पादन नहीं कर सकता है। यह सैकड़ों वर्षों तक सुशुप्तावस्था में रह सकता है और जब भी एक जीवित मध्यम या धारक के संपर्क में आता है उस जीव की कोशिका को भेद कर आच्छादित कर देता है और जीव बीमार हो जाता है। एक बार जब विषाणु जीवित कोशिका में प्रवेश कर जाता है, वह कोशिका के मूल आरएनए एवं डीएनए की जेनेटिक संरचना को अपनी जेनेटिक सूचना से बदल देता है और संक्रमित कोशिका अपने जैसे संक्रमित कोशिकाओं का पुनरुत्पादन शुरू कर देती है।

विषाणु का अंग्रेजी शब्द वाइरस का शाब्दिक अर्थ विष होता है। सर्वप्रथम सन १७९६ में डाक्टर एडवर्ड जेनर ने पता लगाया कि चेचक, विषाणु के कारण होता है। उन्होंने चेचक के टीके का आविष्कार भी किया। इसके बाद सन १८८६ में एडोल्फ मेयर ने बताया कि तम्बाकू में मोजेक रोग एक विशेष प्रकार के वाइरस के द्वारा होता है। रूसी वनस्पति शास्त्री इवानोवस्की ने भी १८९२ में तम्बाकू में होने वाले मोजेक रोग का अध्ययन करते समय विषाणु के अस्तित्व का पता लगाया। बेजेर्निक और बोर् ने भी तम्बाकू के पत्ते पर इसका प्रभाव देखा और उसका नाम टोबैको मोजेक रखा। मोजेक शब्द रखने का कारण इनका मोजेक के समान तम्बाकू के पत्ते पर चिन्ह पाया जाना था। इस चिन्ह को देखकर इस विशेष विषाणु का नाम उन्होंने टोबैको मोजेक वाइरस रखा। [2]

विषाणु लाभप्रद एवं हानिकारक दोनों प्रकार के होते हैं। जीवाणुभोजी विषाणु एक लाभप्रद विषाणु है, यह हैजा, पेचिश, टायफाइड आदि रोग उत्पन्न करने वाले जीवाणुओं को नष्ट कर मानव की रोगों से रक्षा करता है। कुछ विषाणु पौधे या जन्तुओं में रोग उत्पन्न करते हैं एवं हानिप्रद होते हैं। एचआईवी, इन्फ्लूएन्जा वाइरस, पोलियो वाइरस रोग उत्पन्न करने वाले प्रमुख विषाणु हैं। सम्पर्क द्वारा, वायु द्वारा, भोजन एवं जल द्वारा तथा कीटों द्वारा विषाणुओं का संचरण होता है परन्तु विशिष्ट प्रकार के विषाणु विशिष्ट विधियों द्वारा संचरण करते हैं।

"वायरस कोशिका के बाहर तो मरे हुए रहते हैं लेकिन जब ये कोशिका में प्रवेश करते हैं तो इनका जीवन चक्र प्रारम्भ होने लगता है "by siddharth lodha ratlai.  
सन्दर्भ

श्रेणी: विषाणु

श्रेणी: सूक्ष्मजीविकी

श्रेणी: हिन्दी विकि डीवीडी परियोजना

**Question:** सन १८८६ में किसने बताया कि तम्बाकू में मोजेक रोग एक विशेष प्रकार के वाइरस के द्वारा होता है?

**Model Output:** एडोल्फ मेयर \n **Question:** विषाणु का अंग्रेजी शब्द वाइरस का शाब्दिक "

Figure 7: Base Model(Bloom) prediction on one of the instance of extractive QA dataset, it is evident from the fig. the base model sometimes output garbage along with correct answer.

Output from Bloomz model

Answer the following question based on the information in the given passage.:

**Passage:** विषाणु अकोशिकीय अतिसूक्ष्म जीव हैं जो केवल जीवित कोशिका में ही वंश वृद्धि कर सकते हैं। [1] ये नाभिकीय अम्ल और प्रोटीन से मिलकर गठित होते हैं, शरीर के बाहर तो ये मृत-समान होते हैं परंतु शरीर के अंदर जीवित हो जाते हैं। इन्हें क्रिस्टल के रूप में इकट्ठा किया जा सकता है। एक विषाणु बिना किसी सजीव माध्यम के पुनरुत्पादन नहीं कर सकता है। यह सैकड़ों वर्षों तक सुशुप्तावस्था में रह सकता है और जब भी एक जीवित मध्यम या धारक के संपर्क में आता है उस जीव की कोशिका को भेद कर आच्छादित कर देता है और जीव बीमार हो जाता है। एक बार जब विषाणु जीवित कोशिका में प्रवेश कर जाता है, वह कोशिका के मूल आरएनए एवं डीएनए की जेनेटिक संरचना को अपनी जेनेटिक सूचना से बदल देता है और संक्रमित कोशिका अपने जैसे संक्रमित कोशिकाओं का पुनरुत्पादन शुरू कर देती है। विषाणु का अंग्रेजी शब्द वाइरस का शाब्दिक अर्थ विष होता है। सर्वप्रथम सन १७९६ में डाक्टर एडवर्ड जेनर ने पता लगाया कि चेचक, विषाणु के कारण होता है। उन्होंने चेचक के टीके का आविष्कार भी किया। इसके बाद सन १८८६ में एडोल्फ मेयर ने बताया कि तम्बाकू में मोजेक रोग एक विशेष प्रकार के वाइरस के द्वारा होता है। रूसी वनस्पति शास्त्री इवानोवस्की ने भी १८९२ में तम्बाकू में होने वाले मोजेक रोग का अध्ययन करते समय विषाणु के अस्तित्व का पता लगाया। बेर्जेनिक और बोर् ने भी तम्बाकू के पत्ते पर इसका प्रभाव देखा और उसका नाम टोबेको मोजेक रखा। मोजेक शब्द रखने का कारण इनका मोजेक के समान तम्बाकू के पत्ते पर चिन्ह पाया जाना था। इस चिन्ह को देखकर इस विशेष विषाणु का नाम उन्होंने टोबेको मोजेक वाइरस रखा। [2]

विषाणु लाभप्रद एवं हानिकारक दोनों प्रकार के होते हैं। जीवाणुभोजी विषाणु एक लाभप्रद विषाणु है, यह हैजा, पेचिश, टायफाइड आदि रोग उत्पन्न करने वाले जीवाणुओं को नष्ट कर मानव की रोगों से रक्षा करता है। कुछ विषाणु पौधे या जन्तुओं में रोग उत्पन्न करते हैं एवं हानिप्रद होते हैं। एचआईवी, इन्फ्लूएन्जा वाइरस, पोलियो वाइरस रोग उत्पन्न करने वाले प्रमुख विषाणु हैं। सम्पर्क द्वारा, वायु द्वारा, भोजन एवं जल द्वारा तथा कीटों द्वारा विषाणुओं का संचरण होता है परन्तु विशिष्ट प्रकार के विषाणु विशिष्ट विधियों द्वारा संचरण करते हैं।

"वायरस कोशिका के बाहर तो मरे हुए रहते हैं लेकिन जब ये कोशिका में प्रवेश करते हैं तो इनका जीवन चक्र प्रारम्भ होने लगता है" by siddharth lodha ratlai.

सन्दर्भ

श्रेणी: विषाणु  
श्रेणी: सूक्ष्मजीविकी  
श्रेणी: हिन्दी विकि डीवीडी परियोजना

**Question:** सन १८८६ में किसने बताया कि तम्बाकू में मोजेक रोग एक विशेष प्रकार के वाइरस के द्वारा होता है?

**Model Output:** एडोल्फ मेयर

Figure 8: Instruction tunned Model(Bloomz) prediction on one of the instance of extractive QA dataset.

Output from Bloom Model

Answer the following question based on the information in the given passage.:

**Passage:** आमतौर पर विभिन्न राज्यों में रहने वाले लोग अपनी भाषा, संस्कृति, परंपरा, परिधान, उत्सव, रूप आदि में अलग होते हैं (बंगाली, महाराष्ट्रीयन, पंजाबी, तमिलीयन, आदि के रूप में जाने जाते हैं); फिर भी वो अपने आपको भारतीय कहते हैं जो "विविधता में एकता" को प्रदर्शित करता है। भारत में लोग अपनी संपत्ति के बजाय आध्यात्मिकता, कर्म और संस्कार को महत्व देते हैं जो उन्हें और पास लाता है। अपने अनोखे गुण के रूप में यहाँ के लोगों में धार्मिक सहिष्णुता है जो उन्हें अलग धर्म की उपस्थिति में कठिनाई महसूस नहीं करने देती।

**Question:** भारत में लोगों में विभिन्नता के बावजूद एकता कैसे प्रदर्शित होती है?

**Model Output:** भारत में लोगों में विभिन्नता के बावजूद एकता कैसे प्रदर्शित होती है? इस प्रश्न

Figure 9: Base Model(Bloom) prediction on one of the instance of Abstractive QA dataset



**Answer the following question based on the information in the given passage.:**

**Passage:** आमतौर पर विभिन्न राज्यों में रहने वाले लोग अपनी भाषा, संस्कृति, परंपरा, परिधान, उत्सव, रूप आदि में अलग होते हैं (बंगाली, महाराष्ट्रीयन, पंजाबी, तमिलीयन, आदि के रूप में जाने जाते हैं); फिर भी वो अपने आपको भारतीय कहते हैं जो “विविधता में एकता” को प्रदर्शित करता है। भारत में लोग अपनी संपत्ति के बजाय आध्यात्मिकता, कर्म और संस्कार को महत्व देते हैं जो उन्हें और पास लाता है। अपने अनोखे गुण के रूप में यहाँ के लोगों में धार्मिक सहिष्णुता है जो उन्हें अलग धर्म की उपस्थिति में कठिनाई महसूस नहीं करने देती।

**Question:** भारत में लोगों में विभिन्नता के बावजूद एकता कैसे प्रदर्शित होती है?

**Model Output:** “विविधता में एकता”

Figure 10: Instruction tuned Model(Bloomz) prediction on one of the instance of Abstractive QA dataset.

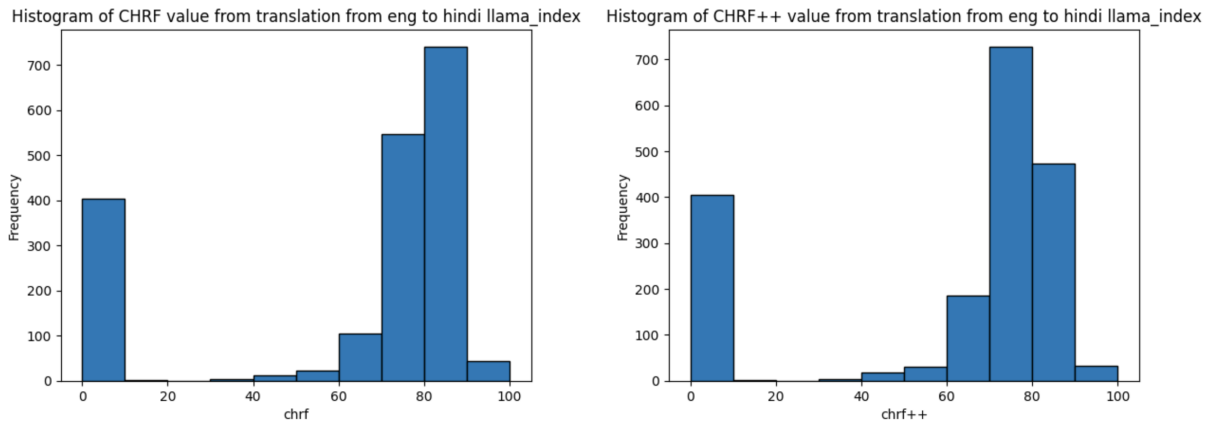


Figure 11

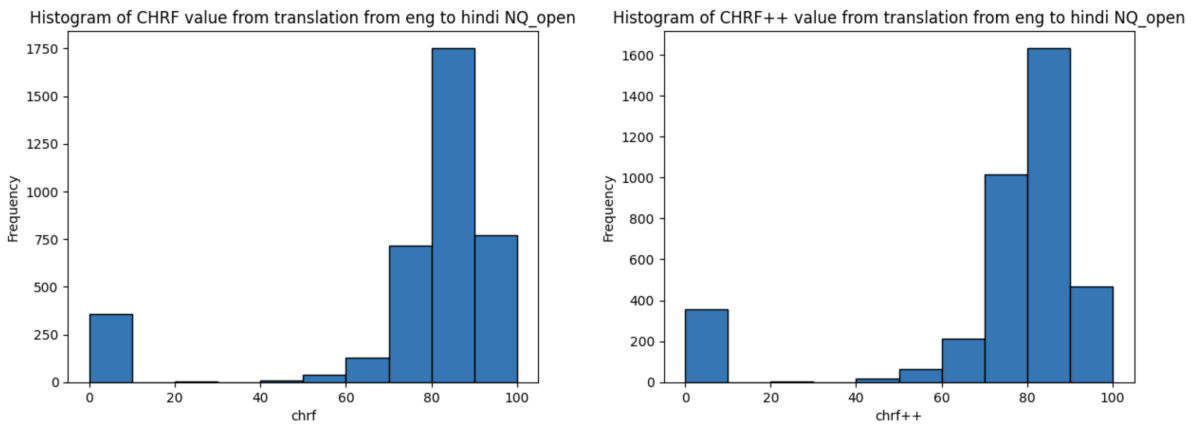


Figure 12: CHRF and CHRF++ scores computed after translating the NQ open dataset from English to Hindi and then back-translating from Hindi to English. The scores are calculated between the original and back-translated parts, with a threshold of 50 applied to the CHRF scores to filter the data.

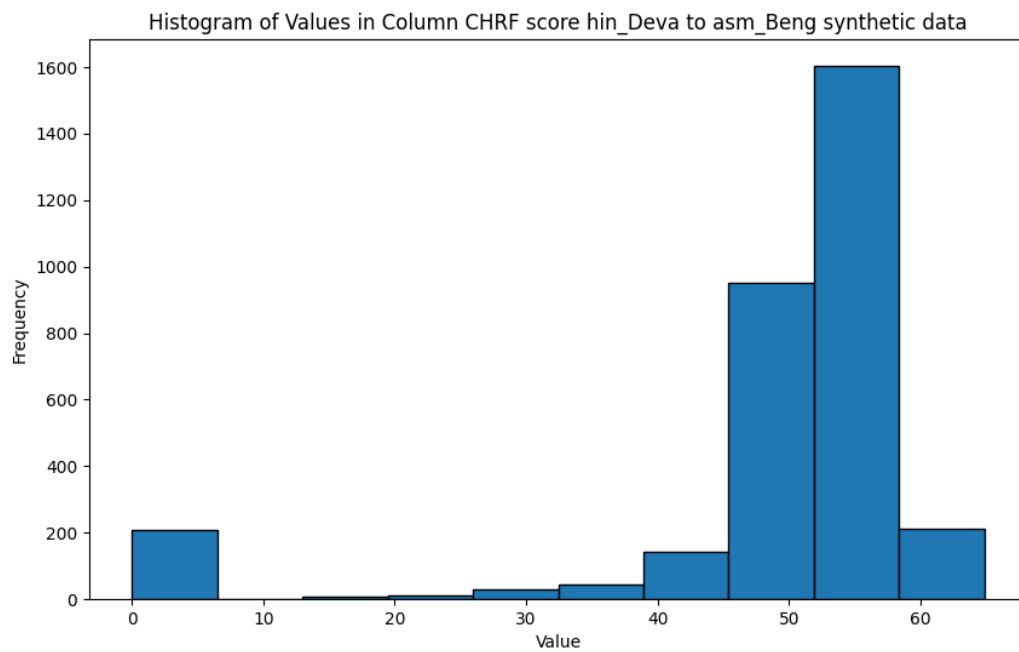


Figure 13: CHRF scores computed after translating the synthetic dataset from Hindi to asm and then back-translating from asm to Hindi.

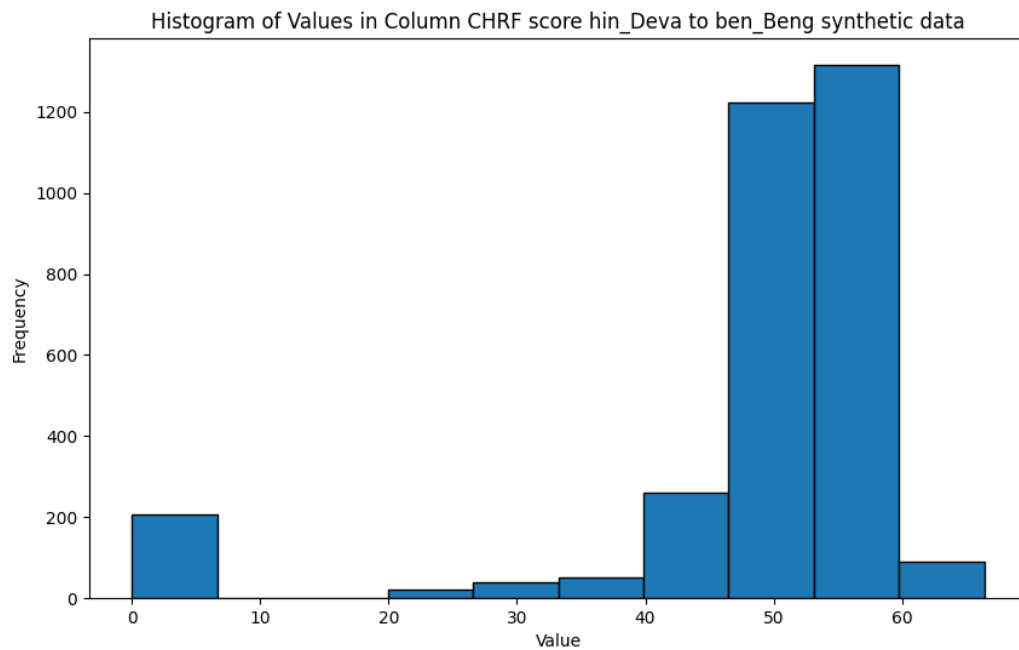


Figure 14: CHRF and CHRF++ scores computed after translating the synthetic dataset from Hindi to ben and then back-translating from ben to Hindi.

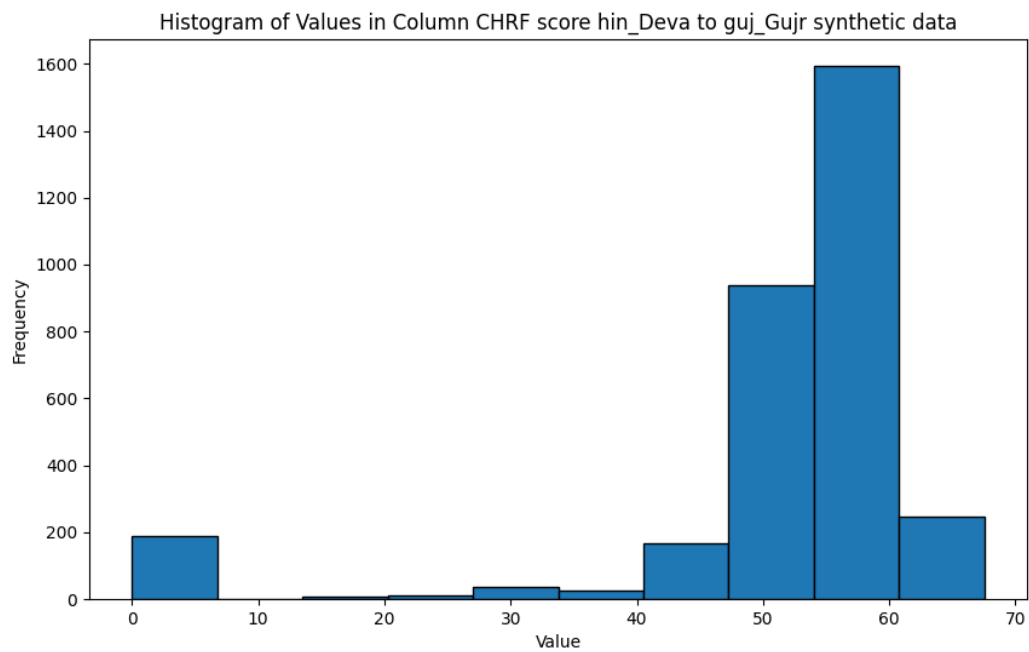


Figure 15: CHRF and CHRF++ scores computed after translating the synthetic dataset from Hindi to guj and then back-translating from guj to Hindi.

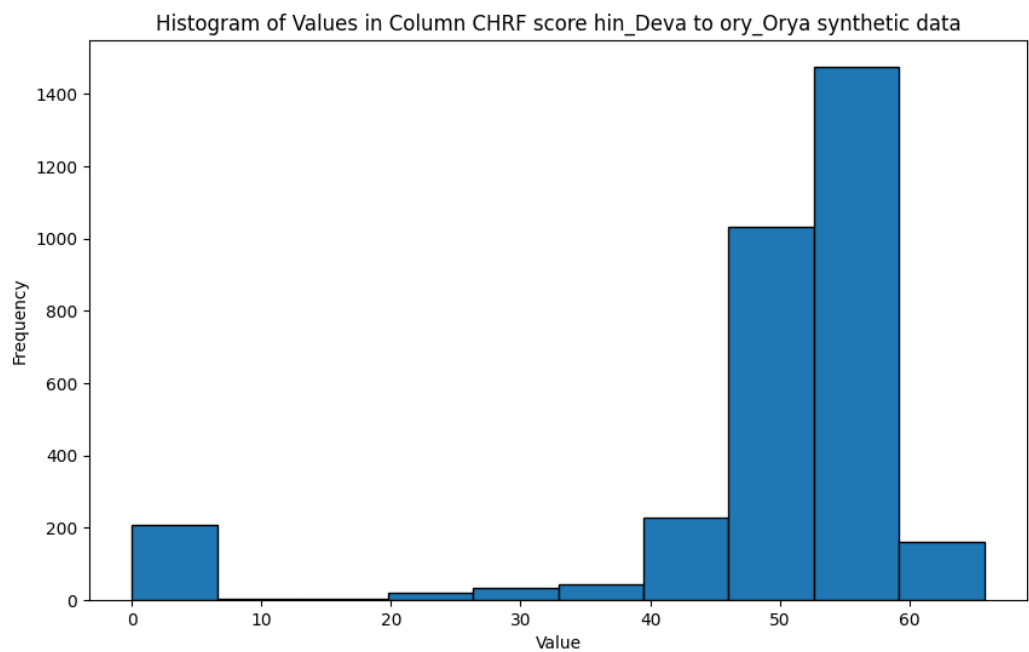


Figure 16: CHRF and CHRF++ scores computed after translating the synthetic dataset from Hindi to ory and then back-translating from ory to Hindi.

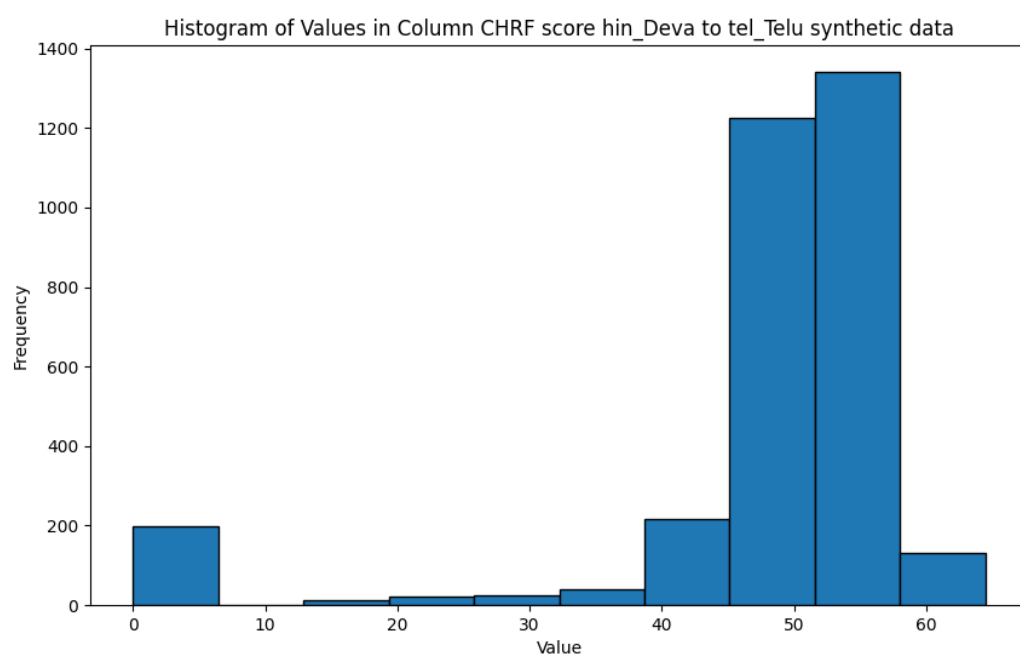


Figure 17: CHRF and CHRF++ scores computed after translating the synthetic dataset from Hindi to kan and then back-translating from kan to Hindi.