MEGA: Multilingual Evaluation of Generative AI

Kabir Ahuja[♡*] **Harshita Diddee**[◇*] **Rishav Hada** [†] **Millicent Ochieng**[†]
Krithika Ramesh [♠*] **Prachi Jain**[†] **Akshay Nambi**[†] **Tanuja Ganu**[†]
Sameer Segal[†] **Maxamed Axmed**[†] **Kalika Bali**[†] **Sunayana Sitaram**[†]
[♡]University of Washington [◇]Carnegie Mellon University
[†]Microsoft Corporation [♠]Johns Hopkins University
kahuja@cs.washington.edu, sunayana.sitaram@microsoft.com

arXiv:2303.12528v4 [cs.CL] 22 Oct 2023

Abstract

Generative AI models have shown impressive performance on many Natural Language Processing tasks such as language understanding, reasoning, and language generation. An important question being asked by the AI community today is about the capabilities and limits of these models, and it is clear that evaluating generative AI is very challenging. Most studies on generative LLMs have been restricted to English and it is unclear how capable these models are at understanding and generating text in other languages. We present the first comprehensive benchmarking of generative LLMs - MEGA, which evaluates models on standard NLP benchmarks, covering 16 NLP datasets across 70 typologically diverse languages. We compare the performance of generative LLMs including Chat-GPT and GPT-4 to State of the Art (SOTA) non-autoregressive models on these tasks to determine how well generative models perform compared to the previous generation of LLMs. We present a thorough analysis of the performance of models across languages and tasks and discuss challenges in improving the performance of generative LLMs on low-resource languages. We create a framework for evaluating generative LLMs in the multilingual setting and provide directions for future progress in the field.

1 Introduction

Large Large Models (LLMs) such as ChatGPT and GPT-4 have created a lot of interest in the AI community and beyond, due to the step jump in their capabilities, such as maintaining context over conversations, fluency of generation, and reasoning. Many users have reported having tested these systems on languages other than English, with varying results, and recent demos of these models (Warren, 2023) have been shown in multiple (albeit high-resource) languages. Recently, the GPT-4 model (OpenAI, 2023) was evaluated on the MMLU multiple choice questions benchmark by automatically translating it into 26 languages, and the results for some low-resource languages in the Latin script were found to be quite promising.

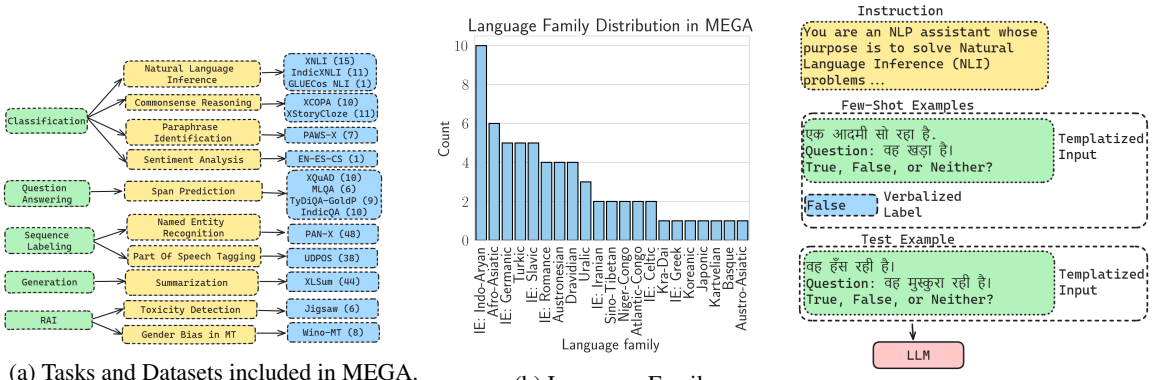
The multilingual capabilities of these models can be traced to their pre-training data, where even the predominantly English large-scale corpora contain hundreds of millions of non-Engish tokens (Blevins and Zettlemoyer, 2022). For GPT-3 unlabeled pre-training data has been documented to contain 119 languages (Brown et al., 2020), where roughly 93% of the tokens are in English[1]. Other LLMs like BLOOM (Scao et al., 2022) and PaLM (Chowdhery et al., 2022) have a better multilingual representation with 60% and 18% non-English data respectively for pre-training. While these models have been trained on multiple languages with varying distributions in the pre-training data, it is not clear how well they perform relative to each other across diverse tasks and languages due to a lack of comprehensive analysis across all models with the same experimental setup.

Recently, there has been a lot of interest in evaluating the different capabilities of LLMs, with comprehensive studies like HELM (Liang et al., 2022) that evaluate these models on a wide variety of capabilities. However, such studies are largely performed on English language data and there is a lack of such large-scale evaluation of LLMs for their multilingual capabilities. Given the current pace at which new language technologies are being developed that use LLMs, the importance of such an evaluation cannot be understated as the cases of inequalities in the performance of previous-generation models across languages have been well-documented (Blasi et al., 2022).

In our work, we present the first large-scale Multilingual Evaluation of Generative AI mod-

[*] Work done when the author was at Microsoft.

[1] https://github.com/openai/gpt-3/blob/master/dataset_statistics/languages_by_word_count.csv



(a) Tasks and Datasets included in MEGA.

(b) Language Family Distribution

(c) Example of multilingual prompting

Figure 1: An overview of our benchmarking exercise: Multilingual Evaluation of Generative AI (MEGA). Numbers in parentheses in Figure 1a contain the number of languages supported in the dataset.

els (MEGA), spanning 16 different datasets, 70 topologically diverse languages, and four LLMs i.e. GPT-3.5 models `text-davinci-003` and `gpt-3.5-turbo`, GPT-4 (`gpt-4-32k`) and BLOOMZ (Muennighoff et al., 2022). We also compare these models with the models fine-tuned on these datasets like TULRv6 (Patra et al., 2022) and MuRIL (Khanuja et al., 2021), which are SoTA on different multilingual benchmarks.

Through our evaluation, we aim to answer three research questions. *(1)*, how well do LLMs fare on multilingual benchmarks compared to fine-tuned SOTA models? *(2)*, what languages do these models perform well in, and can we explain the trends in performance for these models across languages? *(3)*, what prompting strategies should be used for using LLMs for non-English languages?

Our study highlights that there is a significant disparity between the performance of LLMs in English vs non-English languages, especially low-resource languages with non-Latin scripts for which fine-tuned models perform significantly better. While GPT-4 bridges this gap to some extent, the discrepancy still exists. Further, we find that for these languages it is often difficult to do better than simply machine translating the input in a target language to English and then sending it to the LLM for prediction (*translate-test*). We also discuss how different prompt-design choices like prompt-tuning, use of explanations, and number of few-shot examples impact multilingual performance. Finally, we perform some initial analysis to the test the possibility of test data contamination in LLMs that we evaluate and discuss its implications on our findings. Our work provides a blueprint for strategies that can be used for building systems using generative AI for multilingual users. We also release our code [2] for the community to scale up the multilingual evaluation of generative models.

2 MEGA

In this section, we discuss different components of our benchmarking exercise to measure the multilingual capabilities of LLMs. We start by discussing different NLP tasks and datasets that we evaluate these models on, along with their linguistic diversity. We provide an overview of the models we evaluate, baselines for comparison, and describe our evaluation scheme and prompting strategies.

2.1 Datasets and Languages

We broadly consider five families of NLP tasks in our experiments covering 16 different datasets:

Classification Tasks. Here, we further have four different sub-tasks, i) *Natural Language Inference* (classify if a hypothesis is entailed in the premise, contradicts it or neither), which includes XNLI (Conneau et al., 2018) , Indic-XNLI (Aggarwal et al., 2022) (version of XNLI translated to 11 Indian languages), and GLUECos NLI(Khanuja et al., 2020b) for English-Hindi code-mixed data; ii) *Commonsense Reasoning* datasets including causal commonsense reasoning benchmark XCOPA (Ponti et al., 2020) and XStoryCloze (Lin et al., 2022a), where the correct ending of a story with four sentences is to be predicted; iii) *Paraphrase Identification* task PAWS-X (Yang et al., 2019a), where given two sentences, the

[2] https://aka.ms/MEGA

model must predict if the two have the same meaning; iv) EN-ES-CS dataset for *Sentiment Analysis* on English-Spanish code-mixed tweets.

Question Answering (QA). For QA we consider *Span-Prediction* tasks, where the answer to a question is to be predicted within a piece of context provided. We evaluate on XQuAD (Artetxe et al., 2020), MLQA (Lewis et al., 2020), TyDiQA-GoldP (Clark et al., 2020), and IndicQA (Doddapaneni et al., 2022).

Sequence Labeling. This task involves classifying each token in a piece of text and we consider *Named Entity Recognition* dataset PAN-X (Pan et al., 2017) (also called WikiANN) and UDPOS (Nivre et al., 2018) for *Part of Speech Tagging*.

Natural Language Generation (NLG). For NLG we consider the multilingual *Abstractive Summarization* dataset XL-Sum.

Responsible AI (RAI). We consider the multilingual *Toxicity Prediction* dataset Jigsaw(Kivlichan et al., 2020), and Wino-MT to measure *Gender Bias* in MT systems.

All the datasets with the number of languages they include are listed in Figure 1a. These 16 datasets encompass a total of 70 languages covering 21 different language families, with Indo-Aryan and Afro-Asiatic languages in the majority (see Figure 1b). Note that for tasks with $>$ 30 languages i.e. UDPOS, PAN-X, and XL-Sum, we run evaluations on the first 1000 examples of the test sets. For tasks where no public test sets are available (like XQUAD, TyDiQA-GoldP, and IndicQA), we evaluate on validation data. Refer to Appendix §A.1 for a detailed description of all the datasets.

2.2 Models

OpenAI Models. We conduct all benchmarking experiments on the GPT-3.5 models `text-davinci-003` (denoted as DV003 in the paper) and `gpt-3.5-turbo` (Ouyang et al., 2022) (GPT-3.5-Turbo) as well on the GPT-4 model `gpt-4-32k` (OpenAI, 2023). The `text-davinci-003` model has a maximum context size of 4096 tokens, while `gpt-3.5-turbo` and `gpt-4-32k` support context sizes of 16k and 32k respectively.

Baselines. We compare the performance of OpenAI models with two classes of baselines, i) *Prompt-Based baselines*, which like the OpenAI models are evaluated by prompting the model directly for solving a task, and ii) *Fine-tuned Baselines*, which are fine-tuned on task-specific training data. For the former we consider BLOOMZ (Muennighoff et al., 2022), a multi-task fine-tuned version of the BLOOM (Scao et al., 2022) model, which is a 176 billion parameter model trained on 46 natural languages and 13 programming languages. For fine-tuned baselines, we consider TULRv6 (Patra et al., 2022) (the current SoTA on XTREME benchmark), XLMR (Conneau et al., 2020), multilingual BERT (Devlin et al., 2019), and mT5 (Xue et al., 2021). For Indic-datasets we also compare with MuRIL(Khanuja et al., 2021), a multilingual BERT model trained on 16 Indic languages that obtains SOTA performance on many Indic benchmarks. All of these models (excluding mT5 for the XL-Sum and XCOPA), were fine-tuned with English data and then evaluated in a zero-cross-lingual fashion on other target languages.

2.3 Evaluation Methodology

LLMs exhibit two remarkable properties that make them effective at solving a variety of NLP tasks. The first is in-context learning (Brown et al., 2020), where the model learns to solve a task through the few input-output examples provided as part of the context without any weight updates. Secondly, the ability to follow instructions (Mishra et al., 2022; Wei et al., 2021; Ouyang et al., 2022) which is a property of instruction-tuned LLMs, where the models can be prompted to solve new-tasks based on the textual instructions provided in context.

We adopt these two techniques together to test the capabilities of LLMs to solve a variety of tasks in different languages. We define five main components to define the prompts: i) a **test example** x_{test} for which the predictions are to be made; ii) k **few-shot exemplars** $\{(x_i, y_i)\}_{i=1}^{k}$, that are used to provide in-context supervision to the model; iii) a **task instruction** $\mathcal{I}$ which describes the instruction in text for the task to LLM; iv) a **prompt template** $f_{\text{temp}}(x)$ which turns a dataset input example into a text format that can be used for prompting; and v) an **answer verbalizer** $f_{\text{verb}}(y)$ that maps the label y to a textual representation. In our evaluation framework we often consider the instruction, template, and verbalizer as a single entity, and from now on will denote the template to encapsulate the three unless specified separately.

Given these components, the final prompt $f_{\text{prompt}}(x_{\text{test}}; \{(x_i, y_i)\}_{i=1}^{K}, \mathcal{I}, f_{\text{temp}}, f_{\text{verb}})$ or $f_{\text{prompt}}(x_{\text{test}})$ for short for a test input x_{test} can