# In-context Mixing (ICM): Code-mixed Prompts for Multilingual LLMs

**Bhavani Shankar, Preethi Jyothi, Pushpak Bhattacharyya**
Indian Institute of Technology Bombay, India
{bhavanishankar, pjyothi, pb}@cse.iitb.ac.in

## Abstract

We introduce a simple and effective prompting technique called in-context mixing (ICM) for effective in-context learning (ICL) with multilingual large language models (MLLMs). With ICM, we modify the few-shot examples within ICL prompts to be intra-sententially code-mixed by randomly swapping content words in the target languages with their English translations. We observe that ICM prompts yield superior performance in NLP tasks such as disfluency correction, grammar error correction and text simplification that demand a close correspondence between the input and output sequences. Significant improvements are observed mainly for low-resource languages that are under-represented during the pretraining and finetuning of MLLMs. We present an extensive set of experiments to analyze when ICM is effective and what design choices contribute towards its effectiveness. ICM works consistently and significantly better than other prompting techniques across models of varying capacity such as mT0-XXL, BloomZ and GPT-4. Code, prompts and datasets are available here.

## 1 Introduction

In-context learning (ICL) has emerged as a widely accepted gradient-free training paradigm for large language models (LLMs) where a prompt at test-time comprises instructions relevant to the target task, a few task-specific labeled examples and the test instance for which we seek an output (Brown et al., 2020). The few-shot examples in the prompt help the trained model adapt to the task better and elicits more accurate predictions for the test instances.

Creating prompts for ICL with multilingual LLMs (MLLMs) is an emerging area of interest (Shi et al., 2023; Etxaniz et al., 2023; Bang et al., 2023; Huang et al., 2023; Kim et al., 2023). When the test instances are in different target languages, the prompts are typically multilingual with the instructions appearing in English and few-shot examples appearing in the target languages. It is common to use English for instructions in the multilingual prompts as a bridge to the target language, since MLLMs are typically most proficient in English. In this work, we expand further on this intuition of using English as a bridge language.

We introduce a simple and effective prompting technique for MLLMs that we call in-context mixing (ICM). In ICM, the few-shot examples in the target languages are altered to contain intra-sentential code-mixing, i.e., a fraction of content words in the target language sentences are randomly replaced by their English translations. Table 1 shows an example of an ICM prompt when the test instance is in Hindi. We demonstrate the utility of this simple strategy across multiple NLP tasks such as disfluency correction (DC), grammar error correction (GEC) and text simplification (TS) and across multiple MLLMs including open-source models such as mT0-XXL, BloomZ and powerful closed models like GPT-4. We make the following two key observations:

1. We observe that ICM significantly benefits local sequence transduction tasks like DC, GEC and TS that demand a close correspondence between the input sentence and the generated output sentence. ICM does not benefit natural language understanding tasks like sentiment analysis and NLI that rely on sentence-level semantics.

2. We observe that ICM tends to boost the performance of tasks in low-resource languages (i.e. languages that do not appear prominently in the pretraining and/or instruction fine-tuning data of the MLLM) compared to high-resource languages.

We show that ICM outperforms many other



*Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4162–4176

prompting techniques and also show it is complementary to techniques such as chain-of-thought prompting (Shi et al., 2023). We provide a very extensive analysis of ICM including how performance varies depending on the nature of mixing, the degree of mixing, the source of English translations for mixing (i.e., from a lexicon or parallel text), the choice of mixing language, etc. While it appears like one could easily construct alternate prompt modifications that include English and can rival ICM (e.g., append English translations to the target sentences, append target language to English word mappings to the target sentences, etc.), we show that none of these variants provide consistent performance gains like ICM does.

In order to tease apart why ICM works better than other prompting techniques, we probe the MLLM representations for disfluency correction by training a simple MLP classifier that takes MLLM encoder representations of the target sentences as input and labels each word as being disfluent or not. The probe trained on ICM representations performs significantly better on the disfluency classification task compared to all prompting techniques.

While prompting has been extensively studied for English (Brown et al., 2020; Zhao et al., 2021; Kojima et al., 2022; Wei et al., 2022; Fu et al., 2023; Wang et al., 2023), prompting multilingual models to enable improved cross-lingual transfer is relatively far less explored. Thus, we think a simple prompting mechanism like ICM that can consistently improve performance across LLMs and across languages that are under-represented in the pretraining/instruction tuning corpora of the multilingual LLM could be of broader interest.

## 2 Related Work

In the context of LLMs, Brown et al. (2020) introduced the paradigm of In-Context Learning (ICL) which means that the models first develop a range of skills at training time and utilize those skills at inference time to adapt to a target task. ICL refers to combining a query example and an instruction together to form a prompt, which is fed into the model for predictive tasks (Radford et al., 2019). A salient feature of ICL, particularly relevant to large-scale models, is its ability to deliver predictions without changing any of the model parameters.

In recent years, prior work in the area of Multilingual Large Language Models (MLLMs) has explored various prompting techniques. Translate-test is a strong baseline in the traditional pre-train/fine-tune paradigm (Ponti et al., 2021; Artetxe et al., 2023). Recent studies show that it is also effective for prompting autoregressive language models (Lin et al., 2022; Shi et al., 2023), as these models exhibit differential performance depending on the input language (Bang et al., 2023). A newer variant of translate-test is called self-translate (Etxaniz et al., 2023) that overcomes the need for an external translation system. by leveraging the intrinsic few-shot translation capabilities inherent in multilingual LLMs. However, these paradigms falter for same-language sequence-to-sequence tasks where the output must closely mirror the input. This is crucial for tasks requiring high input-output correspondance, like Grammar Error Correction (GEC) and Disfluency Correction (DC).

Recent work has also delved into methodologies like Intra-Cross-Lingual transfer (Winata et al., 2021; Ahuja et al., 2023), as well as the Chain-of-Thought (CoT) style of cross-lingual prompting (Shi et al., 2023; Huang et al., 2023). Nonetheless, a recurring observation across these methods is their tendency to predominantly focus on few-shot example composition within a single language. This approach, albeit comprehensive, often overlooks the potential benefits in cross-lingual transfer when languages are mixed. To boost cross-lingual transferability in multilingual models, a few-shot prompting method was introduced where the source and target languages are mixed (Kim et al., 2023). While the latter explore the merits of language mixing, our work is different in that we modify the few-shot examples that appear in the prompt with intra-sentential code-mixing.

## 3 In-context Mixing (ICM)

A prompt typically comprises an instruction for a given task, accompanied by few-shot examples relevant to that task, followed by a test instance. In "Monolingual Prompting" (Sitaram et al., 2023), the instruction is written in English, while the few-shot examples, the test instance, and the output (when applicable) are all presented in the native language. A "Cross-lingual Prompt" (Sitaram et al., 2023) uses English for the instruction and English or a pivot language for the few-shot examples, with the test instance still being in the native language. In both of the above prompting techniques widely in use, note that the test instances always appear in the native language.

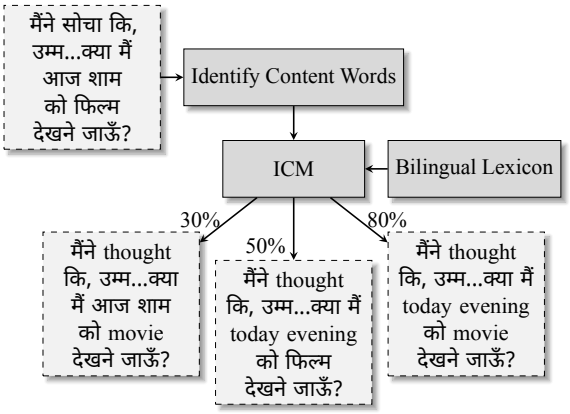


Figure 1: Flowchart illustrating how examples are code-mixed using ICM with varying degrees of mixing.

In-context Mixing (ICM) injects code mixing within the few-shot examples. The examples are now mixed with English, i.e., we randomly switch words in the examples to English. To determine which words are eligible for switching, we restrict our choices to content words with one of the following part-of-speech (POS) tags: noun, verb, adjective, or adverb. Whenever possible, we use bilingual lexicons sourced from Conneau et al. (2017) to facilitate the English replacement, relying on the spaCy tagger (Honnibal et al., 2020) to predict POS tags for each word. For languages lacking these resources, we use NLLB (Costa-jussà et al., 2022) to translate each sentence word by word into English and borrow the English POS tags. We now introduce English words in the example by only considering content words and randomly switching a predetermined fraction of them.

We introduce three different degrees of switching in the examples. We name them "ICM-30", "ICM-50", and "ICM-80". As the name suggests, "ICM-x" means that x% of the content words are randomly switched in each few-shot example. Figure 1 is an illustration of the switching process, and Table 1 shows an example of a "ICM-50" prompt, comparing it with a monolingual prompt. Table 14 in Appendix A shows the prompt template of ICM-50.

It is important to note that switching is done only in the few-shot examples, and the instruction and the test instance are kept intact in their respective languages. In the few-shot examples, switching is introduced both in the input and output of each example. If a word is switched in the input, and that word is also present in the corresponding position in the output, the word in the output is likewise switched. To safeguard against English tokens appearing in the outputs (because the MLLM might learn to also output English target words from the few-shot examples), we explicitly mentioned in the prompt "Do not include English words or vocabulary in the output". Unless specified otherwise, the prompts are 5-shot.

| **Monolingual Prompt** | **ICM-50 Prompt** |
|---|---|
| **Instruction:** Correct disfluencies... (in **English**) | **Instruction:** Correct disfluencies... (in **English**) |
| **Few Shot Examples:** Input: अ मुझे बताइए ये सेवा नहीं है क्या? Output: मुझे बताइए ये सेवा नहीं है क्या? ... (in **native language**) | **Few Shot Examples:** Input: अ मुझे tell ये service नहीं है क्या? Output: मुझे tell ये service नहीं है क्या? ... (**mixed with English**) |
| **Test Instance:** Input: क्या हमें कल.. हमें कल चलना चाहिए। Output: (in **native language**) | **Test Instance:** Input: क्या हमें कल.. हमें कल चलना चाहिए। Output: (in **native language**) |

Table 1: Comparison of the final monolingual prompt (left column) and the final ICM-50 prompt (right column), where 50% of the content words are randomly switched in the few shot examples. Test instances remain in the native language.

## 4 Experimental Setup

### 4.1 Tasks and Datasets

We conduct experiments across five tasks: Disfluency Correction (DC), Grammar Error Correction (GEC), Text Simplification (TS), Natural Language Inference (NLI), and Sentiment Analysis (SA).

#### 4.1.1 Disfluency Correction (DC)

We perform the task of DC on Telugu, Hindi, Marathi, Bengali, Vietnamese, French, and German. For Marathi and Bengali, we use the "real-disfluent" evaluation set from Kundu et al. (2022). We make use of the evaluation sets of the DISCO corpus (Bhat et al., 2023) for Hindi, French and German DC. Vietnamese DC evaluation data was retrieved from Dao et al. (2022). Finally, for Telugu, in the absence of existing datasets, we curated a new evaluation set of 200 disfluent-fluent pairs. A native speaker of Telugu translated 200 English disfluent sentences sourced from Du Bois et al. (2000) and

created corresponding fluent counterparts. Table 18 in Appendix B shows the number of instances for each language.

#### 4.1.2 Grammar Error Correction (GEC)

We present GEC results for Turkish, Korean, German, and Czech. Turkish data comes from Koksal et al. (2020), an evaluation set of erroneous texts from Twitter. For German, we use the Falko subset test set from Boyd (2018), a GEC corpus with Wikipedia edits. For Czech, we use the test set of AKCES-GEC corpus (Náplava and Straka, 2019). Korean evaluation is based on the Korean-Native test set by Yoon et al. (2023). Table 19 in Appendix B details the test instances per language.

#### 4.1.3 Text Simplification (TS)

We study TS for a subset of languages – Brazilian Portuguese, German, and French – from the dataset in Ryan et al. (2023), a collection of 27 resources across 12 languages with over 1.7 million complex-simple sentence pairs. For quicker turnaround, we randomly select 1000 sentences per language for evaluation.

#### 4.1.4 Natural Language Inference (NLI) and Sentiment Analysis (SA)

For NLI, we use test sets of XNLI (Conneau et al., 2018) to evaluate on four languages Hindi, Turkish, French and German. We experiment with SA in Telugu, Hindi, and French. Telugu data comes from the ACTSA Corpus (Mukku and Mamidi, 2017), and we use all 5410 instances for evaluation. For Hindi, we use the test set of IIT Patna Movie review corpus (467 instances) (Akhtar et al., 2016). French SA evaluation uses the evaluation set (655 sentences) from Apidianaki et al. (2016).

### 4.2 Models and Implementation Details

We use the XL(3B) and XXL (11B) variants of the multilingual instruction-tuned LLM mT0-MT, and BloomZ 7B parameter version (Muennighoff et al., 2023). Both these models are mT5 (Xue et al., 2021) and BLOOM (Scao et al., 2022) multitask fine-tuned on xP3mt (Muennighoff et al., 2023), respectively.

P3 (Public Pool of Prompts), (Sanh et al., 2022) an English-only multitask mixture, is a collection of prompted English datasets covering a diverse set of NLP tasks. xP3 is a multilingual version of P3 that integrates datasets from 46 different languages. xP3mt is a collection of the multilingual datasets of xP3 with English and machine-translated prompts. xP3 includes instructions for tasks such as NLI, SA, but not DC, GEC or TS that we mainly evaluate on. BLOOM and mT5 are multitask finetuned on xP3mt to derive mT0-XXL and BloomZ-7B. To check whether our prompting technique works for much larger LLMs, we also show comparisons with the GPT-4 (OpenAI, 2023) version of ChatGPT (OpenAI, 2022) for DC.

We characterize a language to be *low-resource* it is not present in either of pre-training or instruction tuning data, or it is present both during pre-training and instruction-tuning but is less than or equal to 1% or 5% of the overall dataset across all languages, respectively. Based on this characterization, we consider all the languages in this work (except French and German) to be low-resource.

## 5 Main Results

Our baselines include Monolingual, Cross-lingual, Native-CoT and English-CoT prompts. If the steps or "chain-of-thought" is conveyed in the native language of the input and output, it is "Native-CoT", and if expressed in English, it is "English-CoT" (Shi et al., 2023). An example of English-CoT prompt is in Appendix A. CoT is complementary to ICM as a prompting technique. We merge the two to see if we get further gains; ICM30-CoT, ICM50-CoT, and ICM80-CoT are the English-CoT versions of the ICM prompts. That is, the few-shot examples are mixed, and the reasoning appended is in English. An example of ICM50-CoT prompt is in Appendix A. For the tasks NLI and SA, we also include Translate-Test as one of our baselines. That is, we use NLLB to translate both the input sentences and the test instance to English.

Our main observations are summarized below:

1. From Table 2, we see that ICM prompts outperform all the baselines for DC, GEC, and TS, while ICM prompts are not as beneficial for language understanding tasks such as NLI and SA as shown in Table 3. ICM is an effective prompting technique for tasks where a high degree of correspondence between the input and output sequences is critical to maintain. Tasks like DC and GEC retain many of the original input tokens in their output sequences. Semantic understanding tasks like NLI, SA, etc. that do not require such an input-output correspondence and rely on overall sentence semantics do not tend to benefit from ICM.

| | DC | | | | | | | GEC | | | | TS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Prompt** | **Te** | **Hi** | **Mr** | **Bn** | **Vi** | **Fr** | **De** | **Tr** | **Ko** | **Cs** | **De** | **Pt** | **De** | **Fr** |
| Cross-lingual | 9.0 | 10.0 | 3.6 | 3.2 | 5.0 | 21.0 | 10.5 | 8.2 | 10.5 | 13.1 | 6.4 | 29.1 | 30.2 | 32.1 |
| Monolingual | 29.0 | 49.0 | 21.6 | **8.7** | 14.1 | 62.0 | 25.1 | 12.2 | 21.1 | 18.2 | 11.2 | 58.5 | 60.2 | 61.2 |
| Native-CoT | 28.3 | 49.1 | 23.3 | 8.0 | 14.9 | 64.1 | 26.2 | 12.2 | **21.5** | 18.2 | 13.6 | 58.6 | 61.5 | **62.7** |
| English-CoT | **30.2** | **51.2** | **23.5** | 8.6 | **15.4** | **65.0** | **28.3** | **12.3** | 21.4 | **18.8** | **15.5** | **58.9** | **62.0** | 62.6 |
| ICM-30 | 27.8 | 50.1 | 20.8 | 8.0 | 15.4 | 61.4 | 21.7 | 12.5 | 21.6 | 18.6 | 9.9 | 58.9 | 59.1 | 60.1 |
| ICM-50 | 33.8 | 51.2 | 25.6 | 9.8 | 15.7 | 59.6 | 22.1 | 13.0 | 25.4 | 18.9 | 9.9 | 60.4 | 61.4 | 58.7 |
| ICM-80 | 32.3 | 46.9 | 19.6 | 8.3 | 15.9 | 61.3 | 26.0 | 13.0 | 23.1 | 18.2 | 11.5 | 58.1 | 61.6 | 58.7 |
| ICM30-CoT | 29.8 | 51.4 | 21.5 | 8.9 | 15.7 | 61.5 | 24.3 | 12.8 | 23.2 | 19.0 | 12.1 | 58.9 | 58.9 | 58.9 |
| ICM50-CoT | **37.2** | **53.5** | **27.1** | **10.4** | **16.9** | 63.1 | 25.1 | **13.2** | **27.2** | **19.7** | 12.4 | **60.5** | 61.0 | 59.1 |
| ICM80-CoT | 36.1 | 50.1 | 21.2 | 9.2 | 16.4 | 64.1 | 28.0 | 13.1 | 25.4 | 19.6 | 13.6 | 58.6 | 61.1 | 59.3 |

Table 2: Results for DC, GEC, and TS tasks using mT0-XXL. We report exact match scores for DC and GEC, while we report SARI scores for text simplification. The best baseline (among Cross-lingual, Monolingual, Native-CoT, English-CoT) and the best ICM results are highlighted in bold. Statistically significant improvements compared to the best baseline (at $p < 0.01$ using the Wilcoxon signed rank test) are highlighted in green.

| | NLI | | | | SA | | |
|---|---|---|---|---|---|---|---|
| **Prompt** | **Hi** | **Tr** | **Fr** | **De** | **Te** | **Hi** | **Fr** |
| Cross-lingual | 53.0 | 52.3 | 57.1 | 57.0 | **50.1** | 52.4 | 58.2 |
| Monolingual | **56.7** | 57.7 | 59.0 | 59.3 | 49.8 | **53.0** | **60.1** |
| Translate-Test | **56.7** | **58.9** | **59.5** | **59.4** | 47.2 | 48.1 | 56.4 |
| ICM-30 | 56.7 | 57.5 | 59.1 | 59.6 | 49.1 | 52.1 | 57.1 |
| ICM-50 | 56.1 | 57.3 | 58.9 | **60.1** | 50.7 | 51.9 | 55.4 |
| ICM-80 | 54.7 | 56.1 | 58.8 | 59.9 | 51.3 | 51.9 | 55.0 |

Table 3: Accuracy results for the tasks NLI and SA, using mT0-XXL. The baselines are Cross-lingual, Monolingual, and Translate-Test prompts. Statistically significant improvements (at $p < 0.01$ using the Wilcoxon signed rank test) are highlighted in green.

| | DC | | | | GEC |
|---|---|---|---|---|---|
| **Prompt** | **Te** | **Hi** | **Mr** | **Bn** | **Tr** |
| Cross-lingual | **56.1** | 52.8 | 43.7 | 38.3 | 54.1 |
| Monolingual | 54.2 | **56.4** | 41.2 | **39.2** | 56.3 |
| English-CoT | 55.5 | 55.6 | **46.4** | 38.8 | **57.7** |
| ICM-30 | 56.3 | 58.8 | 45.4 | 38.5 | 61.1 |
| ICM-50 | **59.5** | **60.6** | **48.6** | **40.7** | **64.4** |
| ICM-80 | 58.6 | 55.3 | 44.3 | 40.1 | 62.5 |

Table 4: Results with ChatGPT (GPT-4 version) for Telugu, Hindi, Bengali, Marathi DC and Turkish GEC. The exact match scores given are the average across three runs. Statistically significant improvements (at $p < 0.01$ using the Wilcoxon signed rank test) over the highest baseline score are highlighted in green.

2. ICM-50 consistently yields scores that are either comparable to or superior to those achieved using monolingual prompts. The degree of mixing does appear to be a useful hyperparameter to tune. ICM-30 is not enough mixing and ICM-80 is too much mixing so as to distort the structure of the target language sentence.

3. ICM tends to benefit low-resource languages more than high resource languages (with some exceptions). ICM shows clear improvements for low-resource Indic languages such as Telugu, Marathi, etc., Vietnamese, Turkish. French and German are examples of high-resource languages that do not benefit from ICM in Table 2.

Results in Table 2 are with mT0-XXL; results with the models mT0-xl and BloomZ-7B follow the same trend and are summarized in Appendix D. To test whether our results scale to larger models, we also show DC experiments using ChatGPT (GPT-4) in Table 4. Compared to mT0-XXL (Table 2), all the baseline numbers significantly improve with up to 30 absolute points for Bengali. Even with the significantly improved baseline numbers, ICM provides consistent gains that are statistically significant (at $p < 0.01$). We use the Wilcoxon signed rank test for all our experiments in this work.

## 6 Ablations and Probing Experiments

### 6.1 Random vs. Natural Switching

Random switching of words, as in ICM, does not result in naturally code-mixed sentences. To check whether introducing natural/realistic mixing in the examples gives more gains compared to random code-mixing, we compare both strategies on DC for Hindi, Telugu, Marathi, and Bengali. The realistic code-mixed examples were generated by native

speakers of these four target languages. From Table 5, we see that natural mixing outperforms monolingual prompts but it is comparable in performance to random switching. Hence, the effectiveness of using natural code-mixed prompts relative to random mixing remains inconclusive. This is encouraging because random switching can be fully automated and does not require any human intervention, unlike the creation of realistic code-mixed examples. An example demonstrating the qualitative difference between a natural and a random mixed sentence is in Appendix E.

| | Te | Hi | Mr | Bn |
|---|---|---|---|---|
| ***Monolingual*** | 29% | 49% | 21.6% | 8.7% |
| **ICM-*50*** | 33.8% | **51.2%** | **25.6%** | 9.8% |
| **ICM-*natural*** | **34.5%** | 50.8% | 23.2% | **11%** |

Table 5: Comparison of exact match scores for Telugu, Hindi, Marathi, and Bengali DC: ICM-50 for random and ICM-Natural for natural switching. The highest score among random and natural is highlighted in bold.

### 6.2 Content word vs. Arbitrary Switching

For ICM, is it important to switch only content words? We perform DC on Telugu and GEC on Turkish with the same ICM strategy, with relaxing the constraint that only content words can be switched. Table 6 shows that prompts with arbitrary switching perform worse than even monolingual prompts. Switching of content words is a critical choice in ICM for performance gains.

| | Telugu (DC) | | Turkish (GEC) | |
|---|---|---|---|---|
| Monolingual | 29.0% | | 12.2% | |
| | **C-W** | **Arbitrary** | **C-W** | **Arbitrary** |
| ICM-30 | 27.8% | 25.5% | 12.5% | 12.1% |
| ICM-50 | **33.8%** | 22.5% | **13.0%** | 11.0% |
| ICM-80 | 32.3% | 23.3% | **13.0%** | 10.3% |

Table 6: Comparison of exact match scores for content-word and arbitrary switching for Telugu DC and Turkish GEC. **C-W** refers to switching only content words. Statistically significant improvements (at $p < 0.01$ using the Wilcoxon signed rank test) compared to Monolingual are highlighted in bold.

### 6.3 Usage of Alignment Tools

We used bilingual lexicons and POS taggers for switching, whenever these resources were available for a target language. For those languages for which at least one of these resources was not available, we adopted a noisier approach of translating the test instance word by word to English, getting its POS tag, and then randomly switching content words. We compare this approach with the usage of existing alignment tools. We use awesome-align (Dou and Neubig, 2021) to align Hindi and Vietnamese sentences with their English translations from Costa-jussà et al. (2022), then randomly switch the words based on POS tags of aligned English words. From Table 7, we observe that there is no significant difference between using an aligner and translating word by word.

| | Vietnamese | | Hindi | |
|---|---|---|---|---|
| Monolingual | 14.1% | | 39.3% | |
| | **wbw** | **align** | **wbw** | **align** |
| ICM-30 | **15.4%** | 15.1% | 40% | 39.8% |
| ICM-50 | **15.7%** | **15.8%** | **42%** | **42.1%** |
| ICM-80 | **15.9%** | **15.6%** | 34.6% | 34.2% |

Table 7: Comparison of Vietnamese and Hindi DC performance with mT0-XXL, when using word by word (wbw) translation, and awesome-align (align) (Dou and Neubig, 2021) for mixing. Statistically significant improvements (at $p < 0.01$ using the Wilcoxon signed rank test) compared to Monolingual are highlighted in bold.

### 6.4 Switching with other Languages

We created ICM prompts with high-resource languages other than English, such as French and Spanish. Following English, these two languages took up the largest fraction of the overall pretraining and multitask finetuning data. We tested ICM with these languages using few-shot examples for Telugu and Hindi DC, Turkish GEC and Brazilian Portuguese TS with mT0-XXL. From Table 8, we observe that while French and Spanish yield minor performance gains, these are not nearly as substantial as when mixing with English, except the case of Pt-br, where mixing with Spanish has comparable results to that of mixing with English. English is disproportionately larger in the pretraining and/or instruction tuning mixture xP3-mt (approx. 40%) compared to French and Spanish (approx. 6% each). This suggests language proportion, similarity, and influence affects ICM's effectiveness.

### 6.5 Switching only in the Input

In all the experiments so far, in the few-shot examples, switching is introduced both in the input and output of each example. Consider a DC example. If

| | Te (DC) | | | Hi (DC) | | | Tr (GEC) | | | Pt-br (TS) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Monolingual | 29% | | | 39.3% | | | 12.2% | | | 58.5 | | |
| | **En** | **Fr** | **Es** | **En** | **Fr** | **Es** | **En** | **Fr** | **Es** | **En** | **Fr** | **Es** |
| ICM-30 | 27.8% | 28.4% | 22.4% | 40.0% | 41.3% | 40.8% | 12.5% | 12.2% | 12.3% | 58.9% | 57.9% | 58.6% |
| ICM-50 | **33.8%** | **33.0%** | 23.1% | **42.0%** | 40.0% | 38.0% | **13.0%** | 12.1% | 12.5% | **60.4%** | 57.1% | **60.2%** |
| ICM-80 | 32.3% | 23.6% | 24.6% | 34.6% | 30.0% | 28.3% | **13.0%** | 11.3% | 12.5% | 58.1% | 56.0% | 58.3% |

Table 8: Comparison of Telugu and Hindi DC, Turkish GEC, and Brazilian Portuguese TS performance with mT0-xxl, when creating ICM prompts by mixing with French and Spanish other than English. Statistically significant improvements (at $p < 0.01$ using the Wilcoxon signed rank test) compared to Monolingual are highlighted in bold.

a word is switched in the input (disfluent sentence), and that word is also present in the corresponding position in the output (fluent sentence), the word in the output is likewise switched. We check the influence of introducing switching exclusively in the input and leaving the output of the few-shot examples unaltered. This is done in order to determine whether it is crucial for the word switched to English in the input to also be switched in the output. We performed Hindi DC and Brazilian Portuguese TS with pairs of few-shot examples being switched only in the input, as shown in Table 9. We observe that maintaining consistency in switching across the input-output pairs is indeed significant.

| | Hi (DC) | | Pt-br (TS) | |
|---|---|---|---|---|
| Mono | 49.0% | | 58.5 | |
| | **Both** | **only Input** | **Both** | **only Input** |
| ICM-30 | 50.1% | 44.3% | 58.9% | 58.3% |
| ICM-50 | **51.2%** | 42.1% | **60.4%** | 57.2% |
| ICM-80 | 46.4% | 38.6% | 58.1% | 56.1% |

Table 9: Comparison of exact match scores (Hi DC) and SARI scores (Pt-br TS) for the cases where the switching is done in both input and output of the few-shot examples (**Both**) and partial switching (**only Input**). Here, Mono stands for Monolingual. Statistically significant improvements (at $p < 0.01$ using the Wilcoxon signed rank test) compared to Monolingual are highlighted in bold.

### 6.6 Word Switching vs "Mappings" appended

In this particular experiment, instead of switching the content words in the few-shot examples, we augment the input by appending a list of mappings between content words and their corresponding English counterparts. These mappings are derived from a bilingual lexicon, and are a list of key-values added after the input in the few-shot examples; the output remains unchanged. The experiment is conducted on Hindi DC and Turkish GEC. The pairs in the list are also randomly selected and comprise content words. For ICM-30, the list comprises 30% of content words, randomly chosen, and paired with English words derived from the bilingual lexicon. The main goal here is to assess the importance of contextualizing switching, as opposed to merely providing "switch" mappings in the form of a dictionary. From Table 10, we observe that it is indeed critical for in-context switching of the words in the sentence, and no performance improvements are observed when providing the mappings separately.

| | Hindi (DC) | | Turkish (GEC) | |
|---|---|---|---|---|
| Mono | 49.0% | | 12.2% | |
| | **Switching** | **Map-List** | **Switching** | **Map-List** |
| ICM-30 | 50.1% | 48.6% | 12.5% | 11.6% |
| ICM-50 | **51.2%** | 48.2% | **13.0%** | 11.2% |
| ICM-80 | 46.4% | 48.1% | **13.0%** | 11.4% |

Table 10: Comparison of Hindi DC and Turkish GEC exact match scores with contextual switching (**Switching**) versus appending "switch" mappings to few-shot example inputs (**Map-List**). Statistically significant improvements (at $p < 0.01$ using the Wilcoxon signed rank test) compared to Monolingual (Mono here) are highlighted in bold. An example Hindi Mappings appended prompt is in Appendix A.

### 6.7 Translations Appended and In-context Translations

Expanding on the previous method of appending mappings to the input in few-shot examples, we append the entire English translation to the native language sentence in the few-shot examples (and refer to it as "Translations Appended"). We source the translations from Costa-jussà et al. (2022). To further test for the importance of in-context cues, we align both the input sentence in the native language and the English translation using awesome-align (Dou and Neubig, 2021) and do an "aligned

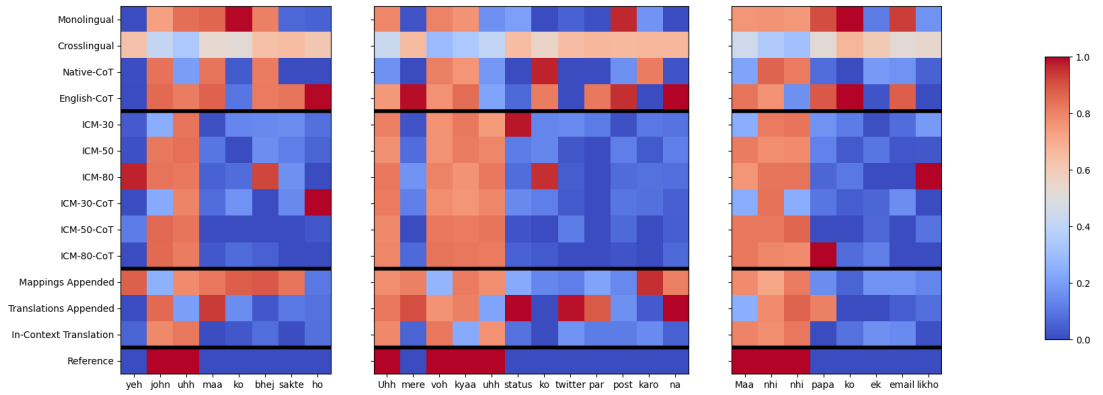


Figure 2: Softmax probabilities of different WC probes for three random Hindi sentences, with disfluent words labeled 1 (red in reference) and fluent words (blue in reference) labeled 0. ICM probes perform the best compared to all counterparts. In-context Translation (as shown in Section 6.7) is the second-best technique.

concatenation" to get a single sequence that alternates between a native language phrase and its corresponding English alignment. We refer to this method as "In-context Translation". We perform these experiments on Telugu and Hindi DC and Korean GEC, as shown in Table 11. Our findings indicate that while simply appending translations does not yield a significant advantage over ICM, In-context Translation improves over the monolingual baseline and narrows the performance gap with ICM. ICM-50 yields significantly better results (at $p < 0.01$ using the Wilcoxon signed rank test) than In-context Translation.

| | Te (DC) | Hi (DC) | Ko (GEC) |
|---|---|---|---|
| Monolingual | 29.0% | 49.0% | 21.1% |
| ICM-30 | 27.8% | 50.1% | 21.6% |
| ICM-50 | **33.8%** | **51.2%** | **25.4%** |
| ICM-80 | 32.3% | 46.9% | 23.1% |
| Translations Appended | 29.1% | 44.9% | 18.9% |
| In-context Translation | 32.5% | 50.3% | 21.5% |

Table 11: Comparison of exact match scores in the case of ICM, Translations Appended, and In-context translations for Telugu and Hindi DC, and Korean GEC. The scores that are significantly better (at $p < 0.01$ using the Wilcoxon signed rank test) than both Monolingual and In-context Translation are highlighted in bold. An example In-context translation prompt is in Appendix A.

### 6.8 Probing Experiments

To probe the representations being learned in the context of varying prompts, we trained a simple single-layer feedforward neural network (FFN) to perform disfluency classification. The training data comprises (last-layer) encoder representations from the model mT0-XXL, for both fluent and disfluent Hindi sentences (3000 sentences each) derived from the training set of PRESTO (Goel et al., 2023). We use a different dataset for hindi here because of lack of fluent sentences in the hindi set sourced from Bhat et al. (2023), which we used consistently for all other experiments.

We train both sentence-level (**SC**) and word-level (**WC**) classifiers. SC uses an FFN probe trained on mean-pooled encoder representations from mT0-XXL to classify the sentence as being disfluent or not, while WC uses an FFN probe that acts on every token within a fixed window of neighbouring words (of size 7) and classifies it as a disfluency or not. Token-level predictions are aggregated through majority voting to get word-level predictions. Evaluations were conducted on disfluent and fluent Hindi sentences in the DISCO corpus (Bhat et al., 2023).

As shown in Table 12, FFN probes trained on representations from sentences within ICM-style prompts outperformed all other techniques in terms of classification accuracy. Figure 2 shows a heatmap of softmax probabilities generated by WC probes for three random Hindi disfluent sentences across all prompting techniques. The visualization clearly highlights that ICM probes are not only accurate, but yield fewer false positives compared to other techniques and demonstrate fairly high confidence in their predictions. In Table 12, we also train SC probes for Korean GEC with grammatically correct and incorrect sentences (1500 each) obtained from Yoon et al. (2023); again, ICM outperforms all other prompting techniques in terms of probe accuracy. The in-context aspect of ICM is important; this is also clear from In-context Translation emerging as the second-best prompting technique. ICM is likely to have an edge over In-context Translation

since the latter would disrupt the overall sentence structure more.

| Prompt | Hi (DC) | | Ko (GEC) |
|---|---|---|---|
| | SC | WC | SC |
| Monolingual | 64.0% | 59.8% | 65.3% |
| Cross-lingual | 32.3% | 18.1% | 45.1% |
| Native-CoT | 66.3% | 61.2% | 68.1% |
| English-CoT | 66.5% | 68.1% | 69.8% |
| ICM-30 | 68.5% | 76.5% | 68.7% |
| ICM-50 | 72.1% | 83.1% | 75.4% |
| ICM-80 | 69.9% | 78.7% | 69.5% |
| ICM-30-CoT | 69.5% | 79.5% | 73.3% |
| ICM-50-CoT | 75.4% | 88.4% | 77.5% |
| ICM-80-CoT | 73.2% | 85.1% | 76.7% |
| Mappings Appended | 63.2% | 64.2% | 66.1% |
| Translations Appended | 66.5% | 67.1% | 66.7% |
| In-context Translation | 69.2% | 75.2% | 70.1% |

Table 12: Comparison of classification accuracies of disfluency detection (Hi) and grammar error detection (Ko) using trained probes at the sentence-level (**SC**) and word-level (**WC**).

## 7 Conclusion

In-Context Mixing (ICM) involving intra-sentential code-mixing has been demonstrated to be an effective prompting technique to use with multilingual LLMs. It significantly improves the performance of NLP tasks that demand a close correspondence between the input and output sequences such as disfluency correction, grammar error correction, and text simplification. This prompting technique particularly benefits low-resource languages such as Telugu, Korean, Turkish, etc. We present extensive ablation experiments and a detailed probing analysis to demonstrate the benefits of ICM across many target languages and models of varying sizes.

## Acknowledgements

The authors thank the anonymous reviewers for their constructive feedback and discussion during the rebuttal that helped improve this submission. The second author would like to gratefully acknowledge a faculty grant from Google Research India supporting her research on multilingual models.

## Limitations

- ICM prompts are not beneficial for natural language understanding tasks like Natural Language Inferencing (NLI) and Sentiment Analysis (SA).
- ICM prompts also tend to work only for low-resource languages (languages that do not appear prominently in the pretraining and/or instruction finetuning data of the MLLM), and not for high-resource languages.

## References

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. 2023. MEGA: Multilingual evaluation of generative AI. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267, Singapore. Association for Computational Linguistics.

Md Shad Akhtar, Ayush Kumar, Asif Ekbal, and Pushpak Bhattacharyya. 2016. A hybrid deep learning architecture for sentiment analysis. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 482–493, Osaka, Japan. The COLING 2016 Organizing Committee.

Marianna Apidianaki, Xavier Tannier, and Cécile Richart. 2016. Datasets for aspect-based sentiment analysis in French. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1122–1126, Portorož, Slovenia. European Language Resources Association (ELRA).

Mikel Artetxe, Vedanuj Goswami, Shruti Bhosale, Angela Fan, and Luke Zettlemoyer. 2023. Revisiting machine translation for cross-lingual classification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6489–6499, Singapore. Association for Computational Linguistics.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718, Nusa Dua, Bali. Association for Computational Linguistics.

Vineet Bhat, Preethi Jyothi, and Pushpak Bhattacharyya. 2023. DISCO: A large scale human annotated corpus for disfluency correction in Indo-European languages. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12833–12857, Singapore. Association for Computational Linguistics.

Adriane Boyd. 2018. Using Wikipedia edits in low resource grammatical error correction. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 79–84, Brussels, Belgium. 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.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

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

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, Loïc 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. *CoRR*, abs/2207.04672.

Mai Hoang Dao, Thinh Hung Truong, and Dat Quoc Nguyen. 2022. Disfluency detection for Vietnamese. In *Proceedings of the Eighth Workshop on Noisy User-generated Text (W-NUT 2022)*, pages 194–200, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

John W Du Bois, Wallace L Chafe, Charles Meyer, Sandra A Thompson, and Nii Martey. 2000. Santa barbara corpus of spoken american english. *CD-ROM. Philadelphia: Linguistic Data Consortium*.

Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lopez de Lacalle, and Mikel Artetxe. 2023. Do multilingual language models think better in english?

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Rahul Goel, Waleed Ammar, Aditya Gupta, Siddharth Vashishtha, Motoki Sano, Faiz Surani, Max Chang, HyunJeong Choe, David Greene, Chuan He, Rattima Nitisaroj, Anna Trukhina, Shachi Paul, Pararth Shah, Rushin Shah, and Zhou Yu. 2023. PRESTO: A multilingual dataset for parsing realistic task-oriented dialogs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10820–10833, Singapore. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Xin Zhao, Ting Song, Yan Xia, and Furu Wei. 2023. Not all languages are created equal in LLMs: Improving multilingual capability by cross-lingual-thought prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12365–12394, Singapore. Association for Computational Linguistics.

Sunkyoung Kim, Dayeon Ki, Yireun Kim, and Jinsik Lee. 2023. Boosting cross-lingual transferability in multilingual models via in-context learning. *CoRR*, abs/2305.15233.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Asiye Tuba Koksal, Ozge Bozal, Emre Yürekli, and Gizem Gezici. 2020. #turki$hTweets: A benchmark dataset for Turkish text correction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4190–4198, Online. Association for Computational Linguistics.

Rohit Kundu, Preethi Jyothi, and Pushpak Bhattacharyya. 2022. Zero-shot disfluency detection for Indian languages. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4442–4454, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Sandeep Sricharan Mukku and Radhika Mamidi. 2017. ACTSA: Annotated corpus for Telugu sentiment analysis. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 54–58, Copenhagen, Denmark. Association for Computational Linguistics.

Jakub Náplava and Milan Straka. 2019. Grammatical error correction in low-resource scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356, Hong Kong, China. Association for Computational Linguistics.

OpenAI. 2022. Chatgpt. https://www.openai.com/.

OpenAI. 2023. Gpt-4 technical report.

Edoardo Maria Ponti, Rahul Aralikatte, Disha Shrivastava, Siva Reddy, and Anders Søgaard. 2021. Minimax and neyman–Pearson meta-learning for outlier languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1245–1260, Online. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, D. Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Michael Ryan, Tarek Naous, and Wei Xu. 2023. Revisiting non-English text simplification: A unified multilingual benchmark. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4898–4927, Toronto, Canada. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2023. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Sunayana Sitaram, Monojit Choudhury, Barun Patra, Vishrav Chaudhary, Kabir Ahuja, and Kalika Bali. 2023. Everything you need to know about multilingual LLMs: Towards fair, performant and reliable models for languages of the world. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 6: Tutorial Abstracts)*, pages 21–26, Toronto, Canada. Association for Computational Linguistics.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 2717–2739. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Genta Indra Winata, Andrea Madotto, Zhaojiang Lin, Rosanne Liu, Jason Yosinski, and Pascale Fung. 2021. Language models are few-shot multilingual learners. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 1–15, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Soyoung Yoon, Sungjoon Park, Gyuwan Kim, Junhee Cho, Kihyo Park, Gyu Tae Kim, Minjoon Seo, and Alice Oh. 2023. Towards standardizing Korean grammatical error correction: Datasets and annotation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6713–6742, Toronto, Canada. Association for Computational Linguistics.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

## A Prompt Examples

| | |
|---|---|
| **Question** | Alice hat 8 Bücher. Sie erhält 3 weitere Bücher von einer Freundin. Wie viele Bücher hat sie jetzt? |
| **Step-by-Step Answer** | Alice starts with 8 books, She receives 3 more books. The total number of books is the sum of the initial and additional books. $8 + 3 = 11$ Therefore, Alice has 11 books now. |
| **Final Answer** | Alice hat jetzt 11 Bücher. |

Table 13: English-CoT Example, where the chain-of-thought is in English, but the other parts of the prompt are in native language (German here).

An example of an English-CoT Prompt is shown in Table 13. ICM-50, ICM50-CoT, Mappings appended, and In-Context Translation prompt examples are shown in table 14, table 15, table 16 and table 17 respectively, for Hindi DC. Similar prompts have been used for other languages.

## B Dataset Instances

Table 18 and Table 19 show the number of instances in each evaluation set for the languages considered for DC and GEC.

## C Number of Few-shot Examples

We performed GEC experiments for Turkish and Czech with mT0-XXL using varying number of few-shot examples (3, 5 or 10) in the prompts. Irrespective of the number of few-shot examples used, we see in Table 20 that ICM prompts consistently surpassed the performance of the baseline prompts (for tasks where ICM prompts are beneficial). Unsurprisingly, increasing the number of examples in the prompt led to improved results for both monolingual and ICM prompts. Even for higher numbers of few-shot examples, at least one variant of the ICM prompts consistently demonstrated superior performance over the monolingual prompts.

## D Results with mT0-xl and BloomZ-7B

Though the trend of mixed Prompts being better than the other counterparts still holds for these models, the corresponding scores for these models are lesser than those obtained with mT0-xxl. mT0-xl because of its smaller size, and BloomZ-7B especially with low resource Indian languages. Table 21 and Table 22 capture these results.

## E Qualitative difference between Random and Natural Mixing

The automatically generated examples with randomly switched content words, at times, yield unnatural generations. Here's an example of a parallel set of natural and randomly code-switched sentences:

- **Natural Mixing:** "मुझे school जाना है" (Switched word: school)
- **Random Mixing:** "मुझे स्कूल go है" (Switched word: जाना to go)

Natural switching might yield मुझे school जाना है (school is the word switched) but when randomly switching, it can be the case that जाना (go, going) can be switched, which cannot switched naturally

| | |
|---|---|
| **Input** | अ मुझे बताइए ये service नहीं है क्या? Remove disfluencies in the given sentence. |
| **Output** | मुझे बताइए ये service नहीं है क्या? |
| **Input** | लेकिन here पर पर अटकने से work नहीं होगा। Remove disfluencies in the given sentence. |
| **Output** | लेकिन here पर अटकने से work नहीं होगा। |
| **Input** | बाजपेयी भी इन अर economists में शामिल थे। Remove disfluencies in the given sentence. |
| **Output** | बाजपेयी भी इन economists में शामिल थे। |
| **Input** | मैंने सोचा कि, उम्म...क्या मैं आज evening को film देखने जाऊँ? Remove disfluencies in the given sentence. |
| **Output** | क्या मैं आज evening को film देखने जाऊँ? |
| **Input** | उसके...मतलब, उसने अपनी new car purchase है। Remove disfluencies in the given sentence. |
| **Output** | उसने अपनी new car purchase है। |
| **Input** | **Test instance** Remove disfluencies in the given sentence. Do not include English words or vocabulary in the output. |
| **Output** | |

Table 14: An example Hindi ICM-50 Prompt for Hindi DC.

| | |
|---|---|
| **Input** | अ मुझे बताइए ये service नहीं है क्या? Remove disfluencies in the given sentence. |
| **Reasoning** | अ मुझे बताइए ये service नहीं है क्या? In this sentence, अ is a filler pause, and hence a disfluency. |
| **Output** | मुझे बताइए ये service नहीं है क्या? |
| **Input** | लेकिन here पर पर अटकने से work नहीं होगा। Remove disfluencies in the given sentence. |
| **Reasoning** | लेकिन here पर पर अटकने से work नहीं होगा। In this sentence, पर is repeated twice, and once instance of it needs to be removed. |
| **Output** | लेकिन here पर अटकने से work नहीं होगा। |
| **Input** | बाजपेयी भी इन अर economists में शामिल थे। Remove disfluencies in the given sentence. |
| **Reasoning** | बाजपेयी भी इन अर economists में शामिल थे। In this sentence, अर has been abruptly corrected to economists, and hence अर is a disfluency and it needs to be removed. |
| **Output** | बाजपेयी भी इन economists में शामिल थे। |
| **Input** | मैंने सोचा कि, उम्म...क्या मैं आज evening को film देखने जाऊँ? Remove disfluencies in the given sentence. |
| **Reasoning** | मैंने सोचा कि, उम्म...क्या मैं आज evening को film देखने जाऊँ? In this sentence, मैंने सोचा कि, is a discourse marker and उम्म... is a filler pause. Both these are disfluencies and they need to be removed. |
| **Output** | क्या मैं आज evening को film देखने जाऊँ? |
| **Input** | उसके...मतलब, उसने अपनी new car purchase है। Remove disfluencies in the given sentence. |
| **Reasoning** | उसके...मतलब, उसने अपनी new car purchase है। In this sentence, उसके... has been corrected to उसने using an edit मतलब. Hence the word that is corrected उसके... and its corresponding edit मतलब are disfluencies. |
| **Output** | उसने अपनी new car purchase है। |
| **Input** | **Test instance** Remove disfluencies in the given sentence. Do not include English words or vocabulary in the output. |
| **Output** | |

Table 15: An example ICM50-CoT prompt for Hindi DC. ICM50-CoT is the En-CoT version of ICM-50 prompt.

in this sentence. The structure of the sentence itself needs to be changed if "go" needs to be used, making it not natural to say मुझे स्कूल go है"

## F Computational Resources

The models mT0-XXL and BloomZ are large. We used a single 80GB GPU in the NVIDIA DGX A100 GPU cluster, to run experiments on both the models simultaneously. We used ChatGPT Subscription for GPT-4 experiments.

| | |
|---|---|
| **Input** | अ मुझे बताइए ये सेवा नहीं है क्या?[{सेवा: service}] Remove disfluencies in the given sentence. |
| **Output** | मुझे बताइए ये सेवा नहीं है क्या? |
| **Input** | लेकिन यहां पर पर अटकने से काम नहीं होगा। [{यहां: here}, {काम: work }] Remove disfluencies in the given sentence. |
| **Output** | लेकिन यहां पर अटकने से काम नहीं होगा। |
| **Input** | बाजपेयी भी इन अर अर्थशास्त्रियों में शामिल थे। [{अर्थशास्त्रियों: economists}] Remove disfluencies in the given sentence. |
| **Output** | बाजपेयी भी इन अर्थशास्त्रियों में शामिल थे। |
| **Input** | मैंने सोचा कि, उम्म...क्या मैं आज शाम को फिल्म देखने जाऊँ? [{शाम: evening}, {फिल्म: film}] Remove disfluencies in the given sentence. |
| **Output** | क्या मैं आज शाम को फिल्म देखने जाऊँ? |
| **Input** | उसके...मतलब, उसने अपनी नई कार खरीदी है।[{ नई: new}. { कार: car}, { खरीदी: purchase}] Remove disfluencies in the given sentence. |
| **Output** | उसने अपनी नई कार खरीदी है। |
| **Input** | **Test instance** Remove disfluencies in the given sentence. |
| **Output** | |

Table 16: An example Mappings appended Prompt for Hindi DC.

| | |
|---|---|
| **Input** | अ Uhh मुझे बताइए tell me ये this सेवा service नहीं है क्या not? Remove disfluencies in the given sentence. |
| **Output** | मुझे बताइए ये सेवा नहीं है क्या? |
| **Input** | लेकिन But यहां here पर पर अटकने stuck से काम work नहीं होगा। would not. Remove disfluencies in the given sentence. |
| **Output** | लेकिन यहां पर अटकने से काम नहीं होगा। |
| **Input** | बाजपेयी Bajpai भी also इन these अर अर्थशास्त्रियों economists में शामिल थे। included. Remove disfluencies in the given sentence. |
| **Output** | बाजपेयी भी इन अर्थशास्त्रियों में शामिल थे। |
| **Input** | मैंने I सोचा कि thought, उम्म... umm... क्या मैं should I आज today शाम को evening फिल्म देखने जाऊँ go see movie? Remove disfluencies in the given sentence. |
| **Output** | क्या मैं आज शाम को फिल्म देखने जाऊँ? |
| **Input** | उसके...मतलब He.. means, उसने he अपनी his नई कार new car खरीदी है। bought. Remove disfluencies in the given sentence. |
| **Output** | उसने अपनी नई कार खरीदी है। |
| **Input** | **Test instance** Remove disfluencies in the given sentence. |
| **Output** | |

Table 17: An example In-Context Translation Prompt for Hindi DC.

| Language | Dataset | Instances |
|---|---|---|
| Telugu | - | 200 |
| Marathi | (Kundu et al., 2022) | 250 |
| Bengali | (Kundu et al., 2022) | 300 |
| Vietnamese | (Dao et al., 2022) | 895 |
| French | (Bhat et al., 2023) | 3005 |
| German | (Bhat et al., 2023) | 3096 |
| Hindi | (Bhat et al., 2023) | 3180 |

Table 18: Number of DC test instances for each language.

| Language | Dataset | Instances |
|---|---|---|
| German | (Boyd, 2018) | 1240 |
| Turkish | (Koksal et al., 2020) | 1970 |
| Czech | (Náplava and Straka, 2019) | 2675 |
| Korean | (Yoon et al., 2023) | 4529 |

Table 19: Number of GEC test instances for each language.

| | Turkish | | | Czech | | |
|---|---|---|---|---|---|---|
| **k-shot** | **3** | **5** | **10** | **3** | **5** | **10** |
| Mono | 10.1% | 12.2% | 13.5% | 16.3% | 18.2% | 22.1% |
| ICM-30 | 10.6% | 12.5% | 13.8% | 16.3% | 18.6% | 22.2% |
| ICM-50 | **11.1%** | **13%** | **14.5%** | **17.1%** | **18.9%** | **23.4%** |
| ICM-80 | 10.3% | **13%** | **14.6%** | 16.6% | 18.2% | **23.1%** |

Table 20: Comparison of 3, 5, 10-shot performance with mT0-XXL, on Turkish and Czech GEC. Significant improvements are in bold. Mono refers to Monolingual.

| | DC | | | | | | | GEC | | | Text Simplification | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Prompt** | **Te** | **Hi** | **Mr** | **Bn** | **Vi** | **Fr** | **De** | **Tr** | **Cs** | **De** | **Pt** | **De** | **Fr** |
| Cross-lingual | 8.2 | 9.3 | 5.6 | 3.2 | 4.2 | 42 | 18.5 | 9.3 | 12.2 | 6 | **36.2** | 19.8 | 30.2 |
| Monolingual | 12 | 17.3 | 6.8 | 5.3 | 8.2 | 37.5 | **22** | 11.3 | 14.5 | 8.2 | 34.3 | 29.3 | 29.8 |
| Native-CoT | 12.3 | 18 | 7.3 | 5.4 | 8.8 | 38.1 | 21.2 | **11.4** | 14.4 | **9.6** | 34.5 | 29.8 | **32.6** |
| English-CoT | **14.1** | **18.2** | **7.5** | **5.6** | **9.4** | **45** | **22** | **11.4** | **14.9** | **9.6** | **36.2** | **30.1** | **32.6** |
| ICM-30 | 9.7 | 18.1 | 6.8 | 4.3 | 8.4 | 35.9 | 21.2 | 11.4 | 14 | 7.8 | 36.1 | 29.5 | 29.1 |
| ICM-50 | 12.7 | 18.2 | 8 | 5.3 | 9.3 | 36.6 | 21.4 | 11.2 | 15 | 7.9 | 38.1 | 29.5 | 28.6 |
| ICM-80 | 9 | 17.6 | 6.8 | 4.3 | 10.1 | 35.3 | 22.7 | 11.2 | 14.2 | 7.5 | 37.2 | 29 | 28.6 |
| ICM30-CoT | 10.2 | 18.4 | 7.5 | 4.9 | 9.1 | 35.5 | 21.3 | 11.4 | 15.1 | 8 | 35.9 | 28.5 | 28.9 |
| ICM50-CoT | **15** | **20.1** | **9.1** | **6.4** | 11.2 | 38.1 | 22.1 | **12.5** | **15.5** | 8.2 | 38.5 | 29.8 | 28.9 |
| ICM80-CoT | 14.8 | 19.8 | 8.2 | 5.2 | **11.4** | 36.1 | 22.5 | **12.5** | **15.5** | 9.2 | **39** | 29.7 | 29.1 |

Table 21: Results for DC, GEC, and TS tasks using mT0-XL. We report exact match scores for DC and GEC, while we report SARI scores for text simplification. The best baseline (among Cross-lingual, Monolingual, Native-CoT, English-CoT) and the best ICM results are highlighted in bold. Statistically significant improvements compared to the best baseline (at $p < 0.01$ using the Wilcoxon signed rank test) are highlighted in green.

| | DC | | | | | | | GEC | | | Text Simplification | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Prompt** | **Te** | **Hi** | **Mr** | **Bn** | **Vi** | **Fr** | **De** | **Tr** | **Cs** | **De** | **Pt** | **De** | **Fr** |
| Cross-lingual | 5.2 | 4.2 | 3.6 | 3.6 | 8.9 | 41.3 | 22.4 | 9 | 16.5 | 8.2 | 35 | 22.1 | 43.1 |
| Monolingual | 12 | 7.3 | **5.6** | 5.4 | 11.2 | 46.2 | **22.8** | 10.1 | 16.9 | 8.5 | 38.1 | 26.3 | 45.7 |
| Native-CoT | **12.1** | 7.9 | 5.3 | 5 | 12.8 | **47.8** | **22.8** | 10.2 | 16.9 | **9.9** | 40.3 | 26.7 | 46.1 |
| English-CoT | **12.1** | **8.2** | **7.5** | **6** | **13.2** | 47.5 | **22.8** | **10.7** | **17.8** | **9.9** | **43.1** | **29.3** | **47.5** |
| ICM-30 | 6 | 8 | 5.3 | 5 | 13.4 | 45.5 | 22.1 | 11.2 | 17.6 | 8.9 | 42.1 | 29.4 | 43.2 |
| ICM-50 | 7.5 | 8.2 | 5.4 | 5.3 | 14.3 | 46.1 | 22.8 | 11.3 | 18.8 | **9.9** | 46.5 | **30** | 41.3 |
| ICM-80 | 7.5 | 8 | 5.3 | 5.3 | 15 | 44.2 | 22.8 | 11 | 18.8 | 9.5 | 45.6 | 29.5 | 40.7 |
| ICM30-CoT | 9 | 8.3 | 5.6 | 6 | 14.8 | 44.1 | 22.7 | 12 | 17.9 | 9.1 | 44 | 29.2 | 44.1 |
| ICM50-CoT | **12.5** | **10.1** | **6** | **7** | **16.1** | 44.3 | **23.8** | **12.8** | **19.1** | 8.9 | **46.9** | 29.9 | 42.8 |
| ICM80-CoT | 12.6 | 8.8 | 5.5 | 6.9 | 15.7 | 46.1 | 23.7 | 12.6 | 19 | 8.6 | 46.1 | 29.4 | 38.7 |

Table 22: Results for DC, GEC, and TS tasks using BloomZ-7B. We report exact match scores for DC and GEC, while we report SARI scores for text simplification. The best baseline (among Cross-lingual, Monolingual, Native-CoT, English-CoT) and the best ICM results are highlighted in bold. Statistically significant improvements compared to the best baseline (at $p < 0.01$ using the Wilcoxon signed rank test) are highlighted in green.