A Three-Pronged Approach to Cross-Lingual Adaptation with Multilingual LLMs

Anonymous EMNLP submission

Abstract

Low-resource languages, by its very definition, tend to be under represented in the pre-training corpora of Large Language Models. In this work, we investigate three low-resource cross-lingual approaches that enable an LLM adapt to tasks in previously unseen languages. `Llama-2` is an LLM where Indic languages, among many other language families, contribute to less than 0.005% of the total 2 trillion token pre-training corpora. In this work, we experiment with the English-dominated `Llama-2` for cross-lingual transfer to three Indic languages, Bengali, Hindi, and Tamil as target languages. We study three approaches for cross-lingual transfer, under ICL and fine-tuning. One, we find that adding additional supervisory signals via a dominant language in the LLM, leads to improvements, both under in-context learning and fine-tuning. Two, adapting the target languages to word reordering may be beneficial under ICL, but its impact diminishes with fine tuning. Finally, continued pre-training in one low-resource language can improve model performance for other related low-resource languages.

1 Introduction

Large language models (LLM; Brown et al., 2020; Touvron et al., 2023; Chowdhery et al., 2022; Mesnard et al., 2024) are known to generalise well across several tasks, including in few shot and zero-shot setups. However, there is limited evidence that shows the ability of these models to generalise to tasks in new languages out of the box, especially to those with which the model has limited exposure to. In this work, we investigate how effectively we can leverage the LLMs for cross lingual transfer, especially for adapting it to low-resource languages.

LLMs typically require tens of billions, if not trillions, of tokens for its pre-training. Now, that is a challenge for majority of the languages in the world. More than 80% of languages in the world are 'left behind' (Joshi et al., 2020), and barely have enough digitised data that matches the requirements for pre-training an LLM from scratch. For instance, the most populous country in the world, India, speaks more than 400 languages[1], with 22 of them recognised as scheduled languages by the Government of India. However, none of these languages contribute to more than 0.005% of the pre-training data of an open-source LLM like `Llama-2` (Touvron et al.,

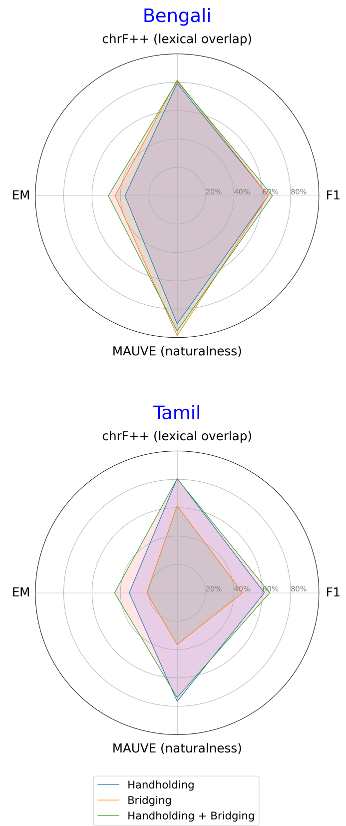

Figure 1: Improved natural language understanding (NLU) and generation (NLG) of `Llama-2-7b` in Bengali and Tamil through continued pre-training in Hindi *(Bridging)* and leveraging English for cross-lingual transfer *(Handholding)*.

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

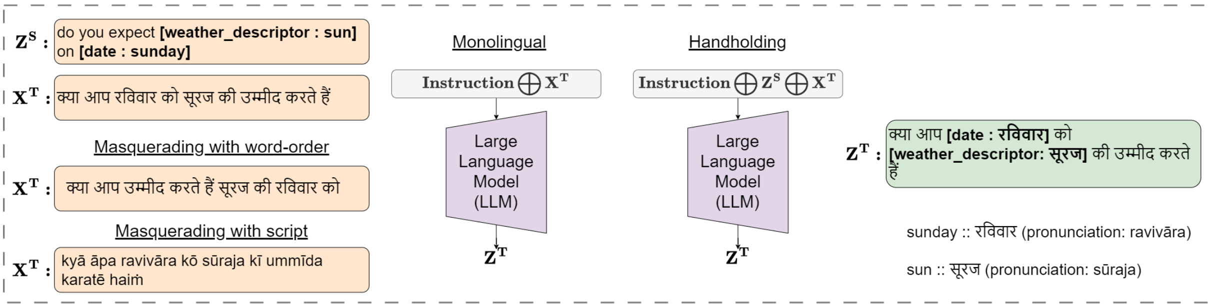

Figure 2: Task of slot filling, using the cross-lingual transfer objective from English to Hindi, using an LLM. In this example, the word 'sun' translates to 'sūraja' in Hindi and 'sunday' translates to 'ravivāra'. Thus, in the output. the LLM assigns the label *weather_descriptor* to the word 'sun' in Hindi, and the label *date* to 'sunday' in Hindi. Refer to Table 11 and Table 12 for details on the prompt.

2023). In fact, more than 95% of these languages lack enough digital resources to incorporate them into an LLM. These resource-poor languages tend to get poorer in representation with the progress in the field (Joshi et al., 2020; Ojo et al., 2024).

Some of the recent works, explore various techniques to adapt an LLM to new languages, especially with limited target language resources (Rathore et al., 2023). Tanwar et al. (2023) exploit cross-lingual transfer to improve in-context learning (ICL) for binary sequence classification tasks in low-resource languages by utilizing in-context exemplars from a high-resource language semantically similar to the input in the target language. Husain et al. (2024) employ continual pre-training on `Llama-2` with romanized pre-training corpora of non-roman script languages, to exploit cross-lingual transfer using the script of English. Awasthi et al. (2023) use 540b PaLM (Chowdhery et al., 2022) to generate training data in low-resource languages using labelled instances in English. Razumovskaia et al. (2024) provide analyses of multilingual capabilities of LLMs on NLU tasks under the settings of in-context learning (ICL), supervised fine-tuning (SFT), and supervised instruction-tuning (SIT).

Our investigation primarily involves the following three questions, centered around information extraction (IE) tasks in a low-resource language using an instruction-tuned LLM. *Q1. Handholding:* For an IE task in a low-resource target language, would providing a parallel, annotated sentence in the predominant language of the LLM, help to exploit cross-lingual transfer, resulting in improved performance for the target language. By predominant language, we imply the language that forms the majority of the pre-training corpora. *Q2. Masquerading:* Would adapting the target language to resemble the predominant language enable in cross-lingual transfer, benefiting the target language. Finally, *Q3. Bridging:* Whether model adaptation in one of the low-resource languages can benefit other related low-resource languages. More clarity on these questions, is presented in Section 2.

We focus on three Indic languages, namely, Bengali, Hindi, and Tamil. These languages are culturally diverse within the Indic context, with Bengali and Hindi belonging to the Indo-Aryan family and Tamil to the Dravidian family. To evaluate our hypotheses *Q1, Q2, and Q3*, we focus on two information extraction tasks: slot filling and named entity recognition (NER). Further, we use a 7 billion parameter English-centric LLM `Llama-2` as our base LLM, unless otherwise stated. The slot filling and named entity recognition tasks possess label-set size of 55 and 3, respectively. Additionally, none of Bengali, Hindi, and Tamil contribute to more than 0.005% of the pre-training corpora of `Llama-2`. Moreover, English is the predominant language, contributing to roughly 90% of the pre-training corpora.

In our experiments, we simlulate a low-resource scenario where we do not expect the target language to have more than roughly $10,000$ instances. In *Bridging*, when `Llama-2` is adapted with Hindi through continued pre-training, we use more than 10,000 sentences in Hindi. However, in this case, Hindi is referred to as the bridge language. The evaluation is solely performed on Bengali and Tamil, both of which satisfy aforementioned criteria for the low-resource setting. Our investigation includes exploiting few-shot in-context learning (ICL) ability of `Llama-2` as well as model adaptation with parameter-efficient supervised fine-tuning (PEFT). To evaluate `Llama-2`, or any auto-regressive LLM in general, we frame the tasks of slot filling and named entity recognition as text-to-text generation tasks. Figure 2 showcases slot filling as a text-to-text generation task.

Extensive experiments on `Llama-2` show that *Handholding* improves NLU and NLG in Bengali, Hindi and Tamil by exploiting cross-lingual transfer from English, under both few-shot ICL and PEFT. Further, *Bridging* with Hindi, improves monolingual task performance in related languages of Bengali and Tamil under PEFT. Ultimately, *Handholding* + *Bridging* turns out the most beneficial combination, yielding best task performance for both low-resource languages of Bengali and Tamil. A quantitative overview has been presented in Figure 1.

Our major contributions can be summarized as follows:

- We demonstrate that the predominant language of an LLM can be leveraged to aid low-resource languages. Specifically, leveraging English via *Handholding*, improves the overall performance of `Llama-2` for information extraction tasks in Hindi, Bengali, and Tamil under both few-shot in-context learning (ICL) and parameter-efficient fine-tuning (PEFT).

- Improved natural language understanding and generation in Bengali and Tamil, as shown by our experiments with `Llama-2` adapted with Hindi *(Bridging)*, demonstrates that adapting a model in one low-resource language can benefit other related languages.

- Modifying target language via *(Masquerading)* to resemble the predominant language, English, gives superficial benefits in few-shot ICL and diminishes further in PEFT.

2 Preliminaries

2.1 Task Definition

Given a finite label-set $\mathcal{L}$, let $\mathbf{X}^S = (X_1^S, X_2^S, \ldots, X_n^S)$ denote a sentence in source language and $\mathbf{A}^S = (A_1^S, A_2^S, \ldots, A_n^S)$ represent the corresponding word-level label sequence, where $A_i^S \in \mathcal{L} \cup \{\phi\}$ and ϕ indicates the absence of a label. A labelled source sequence is given by $\mathbf{Z}^S = ((X_1^S, A_1^S), (X_2^S, A_2^S), \ldots, (X_n^S, A_n^S))$. In *Handholding*, our goal is to transfer these annotations to a parallel, unannotated sentence in target language $\mathbf{X}^T = (X_1^T, X_2^T, \ldots, X_m^T)$, producing an labelled target sentence $\mathbf{Z}^T$. Figure 2 demonstrates the defined text-to-text cross-lingual setup. Formally,

$$\mathbf{Z}^T = \arg\max_{\mathbf{Y}} P_{\text{LLM}}(\mathbf{Y} \mid \mathbf{Z}^S, \mathbf{X}^T)$$

where $\mathbf{Y} = ((Y_1, B_1), (Y_2, B_2), \ldots, (Y_m, B_m))$ is a potential annotated target sentence, with Y_i being elements of $\mathbf{X}^T$ and B_i being elements of $\mathcal{L} \cup \{\phi\}$. In our context, the conditional probability can be decomposed following the auto-regressive nature of LLM generation:

$$P_{\text{LLM}}(\mathbf{Y} \mid \mathbf{Z}^S, \mathbf{X}^T) = \prod_i P((Y_i, B_i) \mid (Y_j, B_j)_{<i}, \mathbf{Z}^S, \mathbf{X}^T)$$

In a similar manner, as shown in Figure 2, a monolingual objective with no *Handholding*, can be formulated in the following manner:

$$\mathbf{Z}^T = \arg\max_{\mathbf{Y}} P_{\text{LLM}}(\mathbf{Y} \mid \mathbf{X}^T)$$

$$P_{\text{LLM}}(\mathbf{Y} \mid \mathbf{X}^T) = \prod_i P((Y_i, B_i) \mid (Y_j, B_j)_{<i}, \mathbf{X}^T)$$

2.2 Handholding, Masquerading, and Bridging

Predominant Language as a Point of Supervision: In our work, with `Llama-2`, English is the predominant language with 89.70% presence in the pre-training corpora of `Llama-2`. On the contrary, low-resource languages like Bengali, Hindi, and Tamil, cover less than 0.005%, and can be regarded as 'unseen' when compared to English. To leverage the understanding of `Llama-2` in English for an IE task in a low-resource 'target' language, we include annotated parallel sentence in English as a part of the task-specific prompt to the LLM. As shown in Figure 2, referred to as *Handholding*, we utilize annotated English sentence ($\mathbf{Z}^{\mathbf{S}}$) to facilitate cross-lingual transfer to the target language.

Adaptation of Target Language: To further aid cross-lingual transfer, we look at ways in which the target language can resemble English. First, we look at word order. Word order refers to the arrangement of words in a sentence. Word order is one of the syntactic features that varies across languages. English follows subject-verb-object order. On the contrary, Indic languages largely follow subject-object-verb word order where the verb appears at the tail part of a sentence. Second, we look at the script of English, to aid cross-lingual transfer. As English follows the Latin script, we employ transliteration schemes to transform the sentence in the target language to Latin. We refer to this adaptation of the target to resemble English as *Masquerading*. Figure 2 gives an overview of target sentence ($\mathbf{X}^\mathbf{T}$) *masqueraded* to resemble English.

Related Language as a Bridge: Continual pre-training (Cui et al., 2024; Gupta et al., 2023), vocabulary extension (Zhao et al., 2024), instruction-tuning(Gala et al., 2024; Li et al., 2023; Husain et al., 2024) are some of the ways to increase representation of language(s) into an LLM. As Hindi is one of the most represented languages in India, we investigate the effect of adapting an LLM in Hindi through continual pre-training, on related low-resource languages of Bengali and Tamil. We refer to this as *Bridging*. Hindi in this scenario, becomes the bridge language, while Bengali and Tamil become the target languages for evaulation.

3 Experiments

3.1 Datasets

Slot Filling: We use Amazon Massive (FitzGerald et al., 2022). The dataset includes slot annotated virtual assistant utterances parallel across 51 languages. We choose sentences from [*utt*] and [*annot_utt*] fields of the dataset to represent unannotated sequence $\mathbf{X}$ and ground-truth annotated sequence $\mathbf{Z}$ respectively for cross-lingual transfer among languages: English, Bengali, Hindi, and Tamil. This dataset includes 55 label types, including `place_name`, `business_name`, `music_genre`, among others. Refer to Table 9 for all label types and Table 8 for the train-test split.

Named Entity Recognition: We work with with AI4Bharat Naamapadam (Mhaske et al., 2023), the largest publicly available NER dataset for 11 Indic languages, sampled and annotated from Samanantar (Ramesh et al., 2022). For the languages in focus, Bengali, Hindi, and Tamil, Naamapadam has 961.7k, 985.8k, and 497.9k instances in their train split, respectively. We sample 16k instances for each of the languages. Due to the absence of ground-truth annotated parallel sequences in English for each of Hindi, Bengali, and Tamil, we leverage the same strategy as (Mhaske et al., 2023) and pick the corresponding set of English sentences from Samanantar and annotate them using a `bert-base` token-classification reference model. List of all label types and train-test split can be found in Table 9 and Table 8, respectively.

3.2 Implementation Details

To evaluate all the hypotheses presented in Section 2, we use English-centric `Llama-2-7b` (Touvron et al., 2023). By ‘English-centric’, we mean to point that English is the predominant language of the LLM. Particularly, we use `Llama-2-7b-chat`, the instruction-tuned variant of pre-trained base `Llama-2-7b`. The need for the instruction-tuned variant is mainly attributed to the nature of a prompt-based generation task where we expect an LLM to be prompted with an instruction followed by an input instance.

For *Handholding*, we use English as the labelled point of supervision to enable cross-lingual transfer. Further, we do not use ground-truth English labels during task-specific model inference; instead, we label the English sentence using a token classification model before the cross-lingual transfer step. We refer to these predicted labels for English as *pseudo* labels and the ground-truth labels for English as *oracle* labels. For slot filling, we use 84.05 F1 score `xlm-roberta-base`[2] token classification model proposed in (Kubis et al., 2023). Whereas, for named entity recognition, we use 91.3 F1 score `bert-base`[3] token classifier, as discussed in Section 3.1. Figure 4 shows the difference between an *oracle* and *pseudo* labelled sentence in English for the task of slot filling.

In *Masquerading* with word order, we use `GIZA++` (Och and Ney, 2003), a word alignment model based on the statistical models by IBM (Brown et al., 1993) and pre-trained LM-based `SimAlign` (Sabet et al., 2021) to generate word re-ordered target sentences. Specifically, we use `SimAlign` for Hindi and `GIZA++` for Bengali and Tamil based on qualitative assessment. In the latter setting of *Masquerading*, we follow `ISO 15919:2001` to transliterate the sentences in Bengali, Hindi, and Tamil to Latin script. Refer Figure 3 for an example of adapting Hindi to resemble English.

For *Bridging*, we utilize `Airavata-7b` (Gala et al., 2024), a continually pre-trained and

[2] `https://huggingface.co/cartesinus/xlm-r-base-amazon-massive-slot`

[3] `https://huggingface.co/dslim/bert-base-NER`

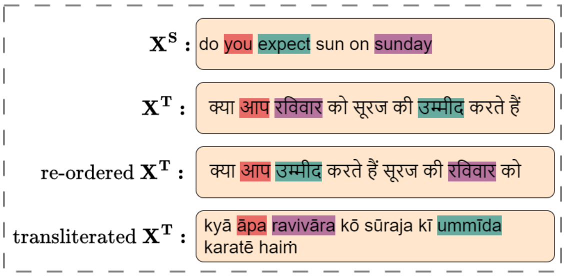

Figure 3: English follows subject verb object word order in contrast to Hindi. Hindi follows the word order of subject object verb As shown, $\mathbf{X^T}$ is presented in SOV order and re-ordered $\mathbf{X^T}$ is presented in SVO order. transliterated $\mathbf{X^T}$ is $\mathbf{X^T}$ in Latin script using `ISO 15919:2001`. Here, only the script of $\mathbf{X^T}$ is changed, keeping the word order of Hindi.

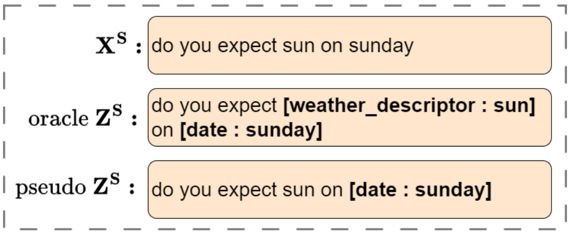

Figure 4: Here, oracle $\mathbf{Z^S}$ refers to the ground-truth annotation of $\mathbf{X^S}$. pseudo $\mathbf{Z^S}$ is obtained after passing $\mathbf{X^S}$ through an `xlm-roberta-base` token classification model.

instruction-tuned version of pre-trained base `Llama-2-7b` model in code-mixed Hindi and English. To ensure that the effect of *Bridging* in Hindi on Bengali and Tamil can be solely attributed to the increased representation of Hindi, we highlight the key differences between `Llama-2-7b-chat` and `Airavata-7b`.

According to Touvron et al. (2023), `Llama-2-7b-chat` builds on `Llama-2-7b` base pre-trained model through supervised fine-tuning with publicly available SFT datasets (Chung et al., 2022) and $27,540$ high-quality in-house vendor-based SFT annotations followed by reinforcement learning through human feedback (RLHF) (Ouyang et al., 2022) with over 1 million human annotated instances. Whereas, to train `Airavata-7b`, Gala et al. (2024) employ LoRA fine-tuning on a continually pre-trained `Llama-2-7b` with publicly available English SFT datasets, with their translations in Hindi, amounting to a total of 385K SFT instances.

We note two observations: (1) the utilized SFT datasets do not cover either of the two datasets used in our evaluation, eliminating any case of labelled data leakage and (2) the quality of the SFT instances used for training `Airavata-7b` does not match that of `Llama-2-7b-chat`, mainly due to absence of high quality in-house annotations and the Hindi subset being translations of publicly available English SFT instances, which generally possess insufficient diversity and insufficient quality (Touvron et al., 2023). Hereafter, we refer to `Llama-2-7b-chat` and `Airavata-7b`, simply as $\texttt{Llama}_{\texttt{chat}}$ and `Airavata`, respectively.

We use HuggingFace `transformers`[4] (Wolf et al., 2020) for task and language adaptation with PEFT and ICL experiments. For ICL, we employ openICL (Wu et al., 2023) and use k-nearest neighbour based retrieval for few-shot demonstrations, following Liu et al. (2022). For retrieval, we compute sentence level representation of the inference time input and the training data using Reimers and Gurevych (2019). We specifically use `xlm-roberta-base` (Conneau et al., 2020) as the base pre-trained model. We choose 8 input-output pairs as for the few-shot demonstrations. These demonstrations for both tasks are mutually exclusive. For instance, in *Masquerading* with word order, we keep all demonstrations to have re-ordered sentences in the target language. It ensures that the few-shot examples are directly relevant to the task variation with high specificity.

For PEFT, we utilize HuggingFace `PEFT`[5] with LoRA (Hu et al., 2021) on top of 4-bit quantization, to fine-tune $\texttt{Llama}_{\texttt{chat}}$ and `Airavata` on a single 80GB NVIDIA A100 Tensor Core GPU. With `PEFT-LoRA`, trainable parameters amount to only 0.5% of the total parameters of the aforementioned LLMs. We train our models with 32-bit paged AdamW (Loshchilov and Hutter, 2019) optimizer, with an initial learning rate of 1×10^{-3} coupled with a *cosine* scheduler. Refer to Appendix D for detailed model configuration.

During inference, we switch to `Contrastive Search`[6] (Su and Collier, 2023) with $\alpha = 0.6$ to penalize token repetitions and control model behavior to generate human-level coherent outputs.

Metrics: We use micro-F1 as our primary evaluation metric for slot filling and named entity recogni-

[4] https://huggingface.co/docs/transformers/index

[5] https://github.com/huggingface/peft

[6] https://huggingface.co/blog/introducing-csearch

tion, both being NLU tasks. Given that both tasks are framed as text-to-text tasks via an LLM, we also include Exact Match to capture correctness, and chrF++ (Popović, 2017) to assess the lexical overlap between the LLM-generated prediction and the ground-truth reference. Additionally, we measure the naturalness of the generated output on 500 randomly sampled test instances using MAUVE (Pillutla et al., 2021).

4 Results

In this section, we present our findings with comparative analysis for the approaches of *Handholding, Masquerading, and Bridging* on `Llama-2` with few-shot ICL and PEFT. For consolidated quantitative figures with PEFT refer to Table 7.

Monolingual ICL Results: We report near zero performance with $\texttt{Llama}_{\texttt{chat}}$ in the monolingual ICL settings. We follow few-shot prompt demonstration under 3 different ICL settings. Here, we provide the input in the target language as is, or *masquerade* it by either transliterating or reordering the input. Nevertheless, we observe near-zero micro-F1, exact match (EM) scores, and poor lexical overlap with reference outputs in all three languages for both the tasks. These observations align with the observations made in (Razumovskaia et al., 2024) and demonstrate the challenges in adapting a new unseen language in ICL settings to an LLM like `Llama-2`.

Language \ Metric	$\texttt{Llama}_{\texttt{chat}}$ *(monolingual)*			
	F1	EM	chrF++	MAUVE
Slot Filling				
Bengali	54.72	22.37	71.40	89.07
Hindi	51.89	23.15	70.90	59.82
Tamil	44.29	14.37	70.65	49.04
Named Entity Recognition				
Bengali	59.98	24.69	85.91	95.28
Hindi	71.58	38.25	90.00	98.70
Tamil	39.92	12.25	68.72	33.06

Table 1: Monolingual performance of $\texttt{Llama}_{\texttt{chat}}$ under PEFT.

Monolingual PEFT Results: As shown in Table 1, we observe performance improvements under monolingual settings, when the model parameters are updated with task-specific PEFT. Averaged over both tasks, the exact match (EM) scores of labelled output generations in Bengali, Hindi, and Tamil stand at 23.53%, 30.7%, and 13.31%, respectively. Whereas, the lexical overlap of the generated outputs with the ground-truth outputs are 78.65%, 80.45%, and 69.68%, respectively. These Indic languages are morphologically rich, in general, leading to lower EM scores, though report higher chrF++ (lexical overlap) and MAUVE (naturalness) scores, comparatively.

Language \ Metric	$\texttt{Llama}_{\texttt{chat}}$ *(Handholding)*			
	F1	EM	chrF++	MAUVE
Slot Filling				
Bengali	64.32	36.82	79.27	90.39
Hindi	60.60	36.70	77.95	89.72
Tamil	61.48	33.79	80.67	76.51
Named Entity Recognition				
Bengali	80.35	45.44	91.00	93.36
Hindi	78.03	47.50	90.38	97.09
Tamil	74.18	42.69	88.75	81.34

Table 2: Effect of *Handholding* on $\texttt{Llama}_{\texttt{chat}}$ under PEFT.

Handholding PEFT Results: Table 2 shows the performance for the target language under PEFT with *Handholding*. We observe that *Handholding* can help further improve the performance in the target language, with task-specific PEFT. Bengali, Hindi and Tamil benefit from labelled sentence in English under PEFT by 9.6%, 8.71%, and 17.19% micro-F1 score for slot filling, and 20.37%, 6.45%, and 34.26% micro-F1 score for named entity recognition. EM scores also improve by an average of 17.6%, 11.4%, and 24.93% for Bengali. Hindi and Tamil, respectively. Similarly, lexical overlap improves in 6 out of 6 cases. However, we observe a drop of 1.92% and 1.61% in naturalness scores of Bengali and Hindi for the NER task.

Language \ Change	*H*	*H+M* (re-ordered)	*H+M* (transliterated)
Slot Filling			
$en_{(source)} \rightarrow bn_{(target)}$	28.02	<u>30.12</u>*	18.01
$en_{(source)} \rightarrow hi_{(target)}$	38.97	<u>40.82</u>*	16.57
$en_{(source)} \rightarrow ta_{(target)}$	22.09	<u>24.38</u>*	12.61
Named Entity Recognition			
$en_{(source)} \rightarrow bn_{(target)}$	13.89	<u>27.88</u>*	17.78
$en_{(source)} \rightarrow hi_{(target)}$	47.61	<u>49.82</u>*	19.61
$en_{(source)} \rightarrow ta_{(target)}$	19.07	<u>30.08</u>*	18.84

Table 3: Micro-F1 scores for the combination of *Handholding (H) and Masquerading (M)* under few-shot ICL.The symbol, * represents statistically significant gains based on pairwise t-tests with just Handholding ($p < 0.05$).

Handholding ICL Results: Similarly, Table 3 reports significant improvements in cross-lingual transfer to the target language when using *Handholding* under ICL settings as well. With few-shot ICL using *Handholding*, we see significant gains, as compared to the near-zero performances with few-shot ICL in monolingual settings. Moreover, we are getting non-zero EM scores in 4 out of 6 cases with *Handholding* under ICL. Nevertheless, as expected, the performance improvements in absolute terms is much higher in *Handholding* with task-specific PEFT (Table 2).

Handholding and Masquerading ICL Results: Further, *Handholding*, along with *Masquerading* via word re-ordering, leads to statistically significant results under ICL. Table 3 shows the results for both *Masquerading* via re-ordering and transliteration. For both the tasks, re-ordering the sentences in all the three languages to resemble the word order in English leads to statistically significant results. However, *Handholding* + *Masquerading* via transliterated target sentences under ICL results in performance drops. As shown in Table 3, the use of transliterated sentences generally results in worse performance than using *Handholding* alone, except for Bengali in NER.

Language \ Change	*H*	*H + M* (re-ordered)
Slot Filling		
$en_{(source)} \rightarrow bn_{(target)}$	<u>64.32</u>	63.19
$en_{(source)} \rightarrow hi_{(target)}$	60.60	<u>61.11</u>
$en_{(source)} \rightarrow ta_{(target)}$	61.48	<u>63.30</u>
Named Entity Recognition		
$en_{(source)} \rightarrow bn_{(target)}$	<u>80.35</u>	55.23
$en_{(source)} \rightarrow hi_{(target)}$	<u>78.03</u>	54.01
$en_{(source)} \rightarrow ta_{(target)}$	<u>74.18</u>	43.96

Table 4: Micro-F1 scores for the combination of *Handholding (H) and Masquerading (M)* under PEFT.

Handholding and Masquerading PEFT Results: As shown in Table 2 and Table 3, *Handholding* benefits the target language, both under ICL and PEFT settings. Similarly, combining *Handholding* with *Masquerading* via word re-ordering has shown to be beneficial under ICL. Table 4 presents the results for the combination of *Handholding* and *Masquerading* with task-specific PEFT. However, the benefits from *Masquerading* appear to diminish or be counterproductive during PEFT, especially for NER tasks. Nevertheless we see statistically significant gains for Slot Filling in Tamil, though not for Hindi. Within *Masquerading*, we do not explore the setting of transliteration of target sentence due to its consistent poor performance under few-shot ICL. For slot filling, Bengali sees a reduction of 1.13% micro-F1 whereas Hindi and Tamil observe increase in micro-F1 scores by 0.51% and 1.82%, respectively.

Language \ Model	$\texttt{Llama}_{\texttt{chat}}$	`Airavata`
Slot Filling		
$bn_{(target)}$	54.72	<u>64.28</u>*
$ta_{(target)}$	44.29	<u>46.03</u>*
Named Entity Recognition		
$bn_{(target)}$	59.98	<u>66.62</u>*
$ta_{(target)}$	39.92	<u>66.14</u>*

Table 5: Micro-F1 scores for the effect of *Bridging* on monolingual performance in Bengali and Tamil. The symbol, * represents statistically significant gains for `Airavata` based on pairwise t-tests with $\texttt{Llama}_{\texttt{chat}}$ ($p < 0.05$).

Bridging: In *Bridging*, Hindi serves as the bridge language, while English still remains the predominant language. In this case, we evaluate model performance on Bengali and Tamil as the target languages. As discussed in Section 3.2, we use `Airavata` to evaluate the effect of increased representation of Hindi on the related languages of Bengali and Tamil. Our first observation follows that *Bridging* improves monolingual performance in both Bengali and Tamil with task-specific PEFT. As shown in Table 5, `Airavata` outperforms $\texttt{Llama}_{\texttt{chat}}$ in both Bengali and Tamil for both tasks of slot filling and named entity recognition. For slot filling, Bengali observes an increase of 9.56% micro-F1, 21.37% increase in EM score, 10.17% increase in lexical overlap and an improved output naturalness by 9.63%. Whereas, Tamil benefits with an increased micro-F1, and EM of 1.74%, and 7.03%. respectively. However, lexical overlap and naturalness of generated outputs with reference outputs falls by 9.31% and 12.52% in Airavata as compared to $\texttt{Llama}_{\texttt{chat}}$. For named entity recognition, we see similar improvements under all metrics, for both languages post *Bridging* except the fall in naturalness for Bengali by 2.47%.

Handholding and Bridging: Table 6 presents the best performing combination, in terms of model performance for slot filling and named entity recog-

Language \ Model	$\texttt{Llama}_{\texttt{chat}}$	`Airavata`
Slot Filling		
$en_{(source)} \rightarrow bn_{(target)}$	64.32	<u>67.21</u>
$en_{(source)} \rightarrow ta_{(target)}$	61.48	<u>65.24</u>
Named Entity Recognition		
$en_{(source)} \rightarrow bn_{(target)}$	80.35	<u>84.80</u>
$en_{(source)} \rightarrow ta_{(target)}$	74.18	<u>82.09</u>

Table 6: Micro-F1 scores for the combination of *Handholding (H)* + *Bridging (B)* under PEFT.

nition. This is achieved by *Bridging* `Llama-2` with Hindi, followed by task-specific model adaptation through PEFT with *Handholding*. In this case, Bengali benefits by 2.89% micro-F1, 11.72% EM score, 1.54% lexical overlap and 4.98% in naturalness as compared to *Handholding* with $\texttt{Llama}_{\texttt{chat}}$ for the task of slot-filling and 4.45% in micro-F1, 13.81% in EM score, 2.86% in lexical overlap and 6.49% in naturalness for named entity recognition. Similarly, for slot filling, Tamil observes increase of 3.84% micro-F1, 10.37% EM score, but a drop in 0.26% lexical overlap and 2.69% naturalness of generated output. Whereas, for named entity recognition, model performance in Tamil increases by 7.91% micro-F1, 19.87% EM score, 5.89% lexical overlap, and 18.12% naturalness score.

5 Conclusion

In this work, through extensive experiments on English-centric `Llama-2-7b-chat` under both ICL and PEFT, we show that *Handholding* improves NLU and NLG in low-resource languages: Bengali, Hindi and Tamil by exploiting cross-lingual transfer from English, demonstrating that the predominant language of an LLM can be leveraged to aid low-resource languages. Further, *Bridging* with a low-resource related language Hindi, results to improved monolingual task performance in related languages of Bengali and Tamil. Ultimately, through *Handholding* + *Bridging*, we show that incorporating both the predominant language of the LLM and adapting the LLM in a related language results to better cross-lingual transfer, leading to significantly improved understanding and generation in other related low-resource languages. However, adapting the target language to resemble the predominant language in terms of syntax and script *(Masquerading)*, only leads to superficial performance improvements in the low-resource language.

Limitations

The very notion of the cross-lingual transfer objective from an labelled sentence in source language to an unannotated sentence in target language requires parallel data. High-quality parallel data is not uniformly available for all language pairs, specifically for underrepresented language families like the Indic family. The requirement of an annotated source during training and/or inference adds up as a bottleneck. As shown in Section 3.2, it can be subdued if we have a reference model to label the source, before cross-lingual transfer. However, the likelihood of a high-accuracy reference model is minimal when considering the case of cross-lingual transfer of annotations between two underrepresented languages.

References

Abhijeet Awasthi, Nitish Gupta, Bidisha Samanta, Shachi Dave, Sunita Sarawagi, and Partha Talukdar. 2023. Bootstrapping multilingual semantic parsers using large language models.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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.

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

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Yiming Cui, Ziqing Yang, and Xin Yao. 2024. Efficient and effective text encoding for chinese llama and alpaca.

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

Jay Gala, 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*.

Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats L. Richter, Quentin Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. 2023. Continual pre-training of large language models: How to (re)warm your model?

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

Jaavid Aktar Husain, Raj Dabre, Aswanth Kumar, Jay Gala, Thanmay Jayakumar, Ratish Puduppully, and Anoop Kunchukuttan. 2024. Romansetu: Efficiently unlocking multilingual capabilities of large language models models via romanization.

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

Marek Kubis, Paweł Skórzewski, Marcin Sowański, and Tomasz Ziętkiewicz. 2023. Back transcription as a method for evaluating robustness of natural language understanding models to speech recognition errors. *arXiv preprint arXiv:2310.16609*.

Haonan Li, Fajri Koto, Minghao Wu, Alham Fikri Aji, and Timothy Baldwin. 2023. Bactrian-x: Multilingual replicable instruction-following models with low-rank adaptation.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali

Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology.

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

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Jessica Ojo, Kelechi Ogueji, Pontus Stenetorp, and David Ifeoluwa Adelani. 2024. How good are large language models on african languages?

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. In *NeurIPS*.

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

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

Vipul Rathore, Rajdeep Dhingra, Parag Singla, and Mausam. 2023. ZGUL: Zero-shot generalization to unseen languages using multi-source ensembling of language adapters. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6969–6987, Singapore. Association for Computational Linguistics.

Evgeniia Razumovskaia, Ivan Vulić, and Anna Korhonen. 2024. Analyzing and adapting large language models for few-shot multilingual nlu: Are we there yet?

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2021. Simalign: High quality word alignments without parallel training data using static and contextualized embeddings.

Yixuan Su and Nigel Collier. 2023. Contrastive search is what you need for neural text generation.

Eshaan Tanwar, Subhabrata Dutta, Manish Borthakur, and Tanmoy Chakraborty. 2023. Multilingual llms are better cross-lingual in-context learners with alignment.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Zhenyu Wu, Yaoxiang Wang, Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Jingjing Xu, and Yu Qiao. 2023. OpenICL: An open-source framework for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 489–498, Toronto, Canada. Association for Computational Linguistics.

Jun Zhao, Zhihao Zhang, Luhui Gao, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Llama beyond english: An empirical study on language capability transfer.

A Evaluation Results

Refer to Table 7 for micro-F1, EM and lexical overlap scores for all experiments with *Handholding*, *Masquerading* and *Bridging* under PEFT.

B Dataset Splits

The dataset split for both tasks is presented in Table 8. For Massive, we use the train, validation, and test split as on HuggingFace `datasets`[7]. For evaluation, we restrict the test set to only contain utterances that have at least 1 token with a slot label. For Naamapadam, we split the 16k sampled instances in a 8:1:1 ratio to create train, validation, and test subsets.

C List of Label Types

Complete list of label types within Massive and Naamapadam is showcased in Table 9.

D Training and Inference Configuration

We present our PEFT and ICL hyperparameter settings in Table 10. These hyperparameters remain the same across both `Llama-2-7b-chat` and `Airavata-7b`.

E Prompt Details

Refer to Tables 11 to 13 for prompts used in our experiments.

[7] https://huggingface.co/datasets/MASSIVE

Language \ Configuration	Llama-2												Airavata							
	monolingual				*H*				*H + M*				*B (monolingual)*				*H + B*			
	F1	EM	chrF++	MAUVE	F1	EM	chrF++	MAUVE	F1	EM	chrF++	MAUVE	F1	EM	chrF++	MAUVE	F1	EM	chrF++	MAUVE
Slot Filling																				
Bengali	54.72	22.37	71.40	89.07	64.32	36.82	79.27	90.39	63.19	0.96	71.81	37.6	64.28	43.74	81.57	98.70	67.21	48.54	80.81	95.37
Hindi	51.89	23.15	70.90	59.82	60.60	36.70	77.95	89.72	61.11	17.29	73.49	24.18	–	–	–	–	–	–	–	–
Tamil	44.29	14.37	70.65	49.04	61.48	33.79	80.67	76.51	63.30	17.80	74.96	19.67	46.03	21.40	61.34	36.52	65.24	44.16	80.41	73.82
Named Entity Recognition																				
Bengali	59.98	24.69	85.91	95.28	80.35	45.44	91.00	93.36	55.23	0.37	54.43	15.14	66.42	34.63	89.45	92.81	84.80	59.25	93.86	99.85
Hindi	71.58	38.25	90.00	98.70	78.03	47.50	90.38	97.09	54.01	0.63	46.18	18.62	–	–	–		–	–	–	–
Tamil	39.92	12.25	68.72	33.06	74.18	42.69	88.75	81.34	43.96	1.31	49.93	45.28	66.14	42.81	91.42	99.22	82.09	62.56	94.64	99.46

Table 7: micro-F1, EM, chrF++, and MAUVE scores under PEFT with the model configurations of *H: Handholding*, *M: Masquerading*, and *B: Bridging*. Here, MAUVE is computed on 500 randomly sampled test instances.

Task \ Dataset Split	Train	Test
Slot Filling	11.5k	1.9k
Named Entity Recognition	12.8k	1.6k

Table 8: Dataset split for slot filling and named entity recognition tasks.

date	time	color_type
house_place	place_name	time_zone
artist_name	timeofday	meal_type
food_type	order_type	news_topic
music_genre	weather_descriptor	playlist_name
device_type	player_setting	song_name
media_type	joke_type	alarm_type
music_descriptor	business_name	business_type
general_frequency	change_amount	event_name
ingredient	person	coffee_type
drink_type	music_album	relation
radio_name	app_name	podcast_descriptor
audiobook_author	audiobook_name	cooking_type
list_name	game_name	podcast_name
movie_type	movie_name	transport_type
transport_name	transport_agency	transport_descriptor
definition_word	currency_name	personal_info
email_address	email_folder	game_type
change_amount		
person (PER)	organization (ORG)	location (LOC)

Table 9: List of all label types in Massive and Naamapadam, in that order.

	Massive	*Naamapadam*
LoRA rank	8	8
LoRA alpha	16	16
Batch size (Training)	32	16
Batch size (Inference)	4	4
Gradient checkpointing	True	True
Gradient accumulation steps	4	4
Max. gradient norm	0.3	0.3
Epochs	2, 3	3
Learning rate	1e-3	1e-3
Optimizer	32-bit AdamW (paged)	32-bit Adam (paged)
Precision	bf16	bf16
LR scheduler	cosine	cosine
Train batch size	32	16
Warm-up ratio	0.05	0.05
Max. sequence length (Training)	512	1024
Stopping Criteria (Inference)	512	768
Penalty alpha (Inference)	0.6	0.6
top_k (Inference)	4	4

Table 10: Complete set of hyperparameters for PEFT and ICL. For ICL, we use the same inference-time hyperparameters as mentioned above.

```
Reinsert the slot annotations into
the following Hindi sentence using the
information in the English sentence.

### Hindi: [Unannotated target]
### English: [Annotated source]
### Output:
```

Table 11: Example prompt format for PEFT with the cross-lingual annotation transfer objective.

```
Reinsert the slot annotations into the
following Hindi sentence.

### Hindi: [Unannotated target]
### Output:
```

Table 12: Prompt format for PEFT with the monolingual annotation objective.

```
«SYS» Add annotations for the
corresponding tokens in Tamil
sentences using the annotation
information given in the English
sentence. The annotations are marked
in the format [annotation_type :
token/value]
Input will be provided in the
following format
### Tamil: Tamil sentence
### English: English sentence
Output should be printed after the
string “### Output:”
The final output should be the Tamil
sentence with annotations inserted
corresponding to the annotations of
the English sentence. Do not add
any extra annotations to the Tamil
sentence, which are not present in
the English sentence input.«/SYS»

Add annotations for the given tokens
<list of tokens present in annotated
source> in Tamil sentence using the
annotation information given in the
English sentence
### Tamil: [Unannotated target]
### English: [Annotated source]
### Output: [Annotated target]
.
.
.
```

$\times$ *n few-shot examples*

```
Add annotations for the given tokens
<list of tokens present in annotated
source> in Tamil sentence using the
annotation information given in the
English sentence
### Tamil: <An unannotated Tamil
sentence>
### English: <An annotated English
sentence>
### Output:
```

Table 13: Example prompt format for few-shot ICL with the cross-lingual annotation transfer objective.