POMP: Probability-driven Meta-graph Prompter for LLMs in Low-resource Unsupervised Neural Machine Translation

Shilong Pan[1], **Zhiliang Tian**[1*], **Liang Ding**[2],
Haoqi Zheng[1], **Zhen Huang**[1], **Zhihua Wen**[1], **Dongsheng Li**[1*]
[1]College of Computer, National University of Defense Technology
[2]JD Explore Academy
{panshilong18, tianzhiliang, zhenghaoqi, huangzhen, zhwen, dsli}@nudt.edu.cn
liangding.liam@gmail.com

Abstract

Low-resource languages (LRLs) face challenges in supervised neural machine translation (NMT) due to limited parallel data, prompting research in unsupervised NMT (UNMT). UNMT, without requiring ground truth, provides solutions for LRL translations using synthetic pseudo-parallel data and parallel data from auxiliary language pairs. However, they usually encounter translation errors, including errors from synthetic data and from auxiliary language pairs with linguistic biases. We argue that large language models (LLMs) mitigate UNMT's translation errors by dynamically organizing auxiliary languages in prompts to improve LRL translations. In this paper, we propose **PrO**bability-driven **M**eta-graph **P**rompter (POMP), an approach employing a dynamic graph to organize multiple auxiliary languages, to prompt LLMs in LRL translations. POMP proposes a language-specific meta-graph that dynamically samples multiple translation paths to organize auxiliary languages in constructing prompts. Following the path, POMP prompts LLMs to translate with a mixture of auxiliary languages. We achieve the meta-graph's evolution by back-propagating evaluation scores to update probabilities on the graph. Our experimental improvements show POMP's effectiveness on LRLs' translation.

1 Introduction

Training supervised NMT models requires extensive parallel data (Xue et al., 2021; Fan et al., 2021), which struggles in low-resource languages (LRLs) that have limited parallel data. Researchers extract (Schwenk et al., 2021a,b) and annotate (Goyal et al., 2022) parallel samples of LRLs, which requires a substantial human effort. Therefore, the research community explores unsupervised methods to achieve NMT without parallel data.

Unsupervised low-resource translation methods are categorized into three types: (1) Back-translation (Sennrich et al., 2016; Lample et al., 2018b) employs existing target-source NMT models to translate target monolingual data to the source language, generating synthetic parallel data for training desired source-target NMT models. This approach is constrained by the translation errors in the synthetic data (Chauhan et al., 2022). (2) Transfer learning-based NMT (Li et al., 2022) train on high-resource auxiliary languages and transfer their ability on LRLs' inference. The linguistic biases between training (auxiliary) and testing languages lead to poor translation accuracy (Dabre et al., 2017). (3) Pivot-based translation (Kim et al., 2019) first translates from the source to a pivot (auxiliary) language and then translates from the pivot to the target. The multi-hop translation process introduces potential translation errors among different languages (Liu et al., 2019). Overall, those methods suffer from translation errors caused by translating among different languages (including synthetic data), where the languages have linguistic biases with each other and are hard to translate, called linguistically biased translation errors.

LLMs greatly improved NMT (Peng et al., 2023) but hardly performed well in LRL since LLMs' pre-training corpora mainly derive from high-resource languages. Some researchers conduct prompt engineering (Diao et al., 2023; Shum et al., 2023; Wen et al., 2024) for LLMs in LRL but achieve limited performance because LLMs learn little for LRLs in zero-shot prompting (Hendy et al., 2023). Researchers apply in-context learning (ICL) that feeds few-shot examples with input-output pairs (source-target parallel pairs) to LLMs and asks LLMs to follow the source-target pairs to translate target sentences (Dong et al., 2023). Alves et al. (2023) enhanced supervised translation by fine-tuning LLMs with parallel sentences. Hendy et al. (2023) found that LLMs still require sufficient parallel data and

*Corresponding Author

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9976–9992
August 11-16, 2024

hardly do well in unsupervised LRLs.

As LLMs have a strong ability to understand high-resource languages, we argue that LLMs with high-resource auxiliary languages should be a good way to assist unsupervised LRL translations, where high-resource languages serve to construct prompts for LLMs. Considering multiple auxiliary languages via LLMs can mitigate linguistically biased translation errors in UNMT.

In this paper, we propose **PrO**bability-driven **M**eta-graph **P**rompter (POMP),[1] which employs a sampling-based dynamic graph organizing auxiliary pseudo-parallel sentences to prompt LLMs to alleviate linguistically biased translation errors in UNMT. Specifically, we design a language-specific meta-graph to generate multiple translation paths to prompt LLMs, where the paths bridge the source and the target, through various auxiliary languages. Multiple paths in the graph prompt LLMs in various ways, which mitigate translation errors caused by an individual language. Using the paths from the meta-graph, we propose two operations to prompt LLMs: (1) *Generate* prompts LLMs with a single auxiliary language along the translation paths to make full use of the selected auxiliary languages; (2) *Aggregate* prompts LLMs with all auxiliary in the path and the best *Generate*'s output to obtain translations considering cross-linguistic information from multiple auxiliary languages. To optimize the meta-graph, we design a probabilistic backward graph evolution strategy that backpropagates the evaluation score for the translation path into each auxiliary language according to the contribution of each language.

Our contributions are as follows: (1) We propose a UNMT that mitigates linguistically biased translation errors using auxiliary languages in translations to achieve SOTA performance on four LRL translations. (2) We propose a prompting graph for LLMs to organize multiple auxiliary languages. (3) We design a probabilistic backward graph evolution algorithm that iteratively updates probabilities of auxiliary languages to construct better prompts.

2 Related Work

2.1 Unsupervised NMT on Low-resource Languages

UNMT methods aim to learn an NMT model without parallel data, offering more practical alternatives for LRLs. UNMT methods in recent years are mainly divided into three categories: back-translation, transfer learning, and pivot-based translation.

Back-translation (Sennrich et al., 2016) uses monolingual data to train NMT models by generating synthetic source sentences. Edunov et al. (2018) confirmed its improved performance in both large-scale and low-resource contexts. Lample et al. (2018a); Artetxe et al. (2018) pioneered UNMT using iterative back-translation, mitigating errors of initial NMT models. This method has been effectively applied to LRLs (Chen et al., 2020; Sánchez-Martínez et al., 2020). However, Edman et al. (2020) noted that poor initial quality in word embeddings and cross-lingual alignments might reduce translation performance in LRLs.

Transfer learning employs a model, trained on high-resource languages, to infer LRLs' translations (Zoph et al., 2016). Chronopoulou et al. (2021) presented a meta-learning algorithm to improve UNMT models in low-resource domains by leveraging knowledge learned from high-resource domains, using slight training data to quickly adapt to new domains. Moreover, Li et al. (2022) continuously transferred knowledge from a high-resource parent model to a low-resource child model during training, ensuring prediction consistency between them.

Pivot-based translation (Leng et al., 2019) bridges from the source to intermediary languages and then onto the target language, facilitating translation when parallel datasets are insufficient. Kim et al. (2019) applied this concept in transfer learning, using pivot languages and parallel corpora for better translations. Currey and Heafield (2019) leveraged monolingual pivot language data to create pseudo-parallel corpora, augmenting data for training. Improper pivot languages chosen in these methods hinder translation results (Liu et al., 2019). Our approach incorporates a dynamic prompting graph to organize auxiliary languages for efficiently prompting LLMs.

2.2 Neural Machine Translation with LLMs

The community has explored various LLMs (Touvron et al., 2023a,b; OpenAI, 2022, 2023), which show promised performance in NMT (Jiao et al., 2023; Hendy et al., 2023; Zhu et al., 2023). Research exploring the translation capabilities of LLMs often involves ICL and fine-tuning methods.

As for ICL methods, Brown et al. (2020) explored the capabilities of LLMs to learn target

[1] Our code is at github.com/slpanir/POMP.

tasks with the prompt made up of in-context exemplars and templates. Garcia et al. (2023) showed comparable performances of ICL to those large, supervised models. Vilar et al. (2023); Agrawal et al. (2023) evaluated various strategies for selecting translation examples for ICL, emphasizing the importance of example quality.

As for fine-tuning methods, (Li et al., 2023) explored enhancing translation by fine-tuning XGLM-7B (Lin et al., 2022), with translation instructions, especially for low-resource languages. Research (Alves et al., 2023) has shown that fine-tuning methods perform better than ICL in LRL translation but at a cost of high computational resources. In contrast, our work integrates two approaches by (1) training a sampling-based graph efficiently compared to billions of parameters, and (2) employing this graph to strategically organize auxiliary languages and construct prompts in ICL.

3 Methods

3.1 Overview

The proposed model, POMP (Fig. 1), consists of four modules: (1) **UNMT-based Pseudo-parallel Generator** (§3.2) generates pseudo-parallel sentences for both source-target and source-auxiliary pairs, where the auxiliary languages assist translation in LLMs' prompts. (2) **Language-specific Meta-graph** (§3.3) carries relations among source, target, and auxiliary languages, which generates multiple translation paths to prompt LLMs. (3) **Graph-Prompting LLM-based Translator** (§3.4) generates multiple translation paths from the meta-graph by sampling to prompt LLMs for translation. (4) **Probabilistic Backward Graph Evolution** (§3.5) updates the probabilities for paths in the meta-graph.

Pseudo-parallel sentences generated in §3.2 are used to calculate probability weights in §3.3, incorporated as prompts in §3.4. The meta-graph in §3.3 samples translation paths, which involve prompting LLMs in LRL translations in §3.4. The evaluation scores of the translations are back-propagated in §3.5 to update probabilities in the meta-graph.

3.2 UNMT-based Pseudo-parallel Generator

Following (Chen et al., 2022), we build a UNMT model to generate pseudo-parallel sentences for both source-target and source-auxiliary language pairs, where the auxiliary languages are extra languages assisting in prompting LLMs. We initialize the transformer-based NMT model with weights of a multilingual pre-trained XLM-R model (Conneau et al., 2020) and then train with a two-stage training method on six auxiliary language datasets.

In the first stage, to preserve the cross-lingual transferability of the encoder, we train the decoder with the auxiliary-English pairs. In the second stage, to further improve the model learning from the training data, we jointly optimize all parameters of the encoder and decoder. Our empirical observation shows this processing transfers its translation ability learned from auxiliary languages in training to LRLs in testing.

We utilize the UNMT model to generate the translations of the LRL source as inputs in §3.4. Then we generate pseudo-parallel sentences of source-auxiliary pairs to construct the meta-graph in §3.3 and prompts in §3.4.

3.3 Language-specific Meta-graph

We design a language-specific meta-graph to explore diverse translation paths with high-resource auxiliary languages to prompt LLMs for LRL translations. We define the meta-graph as $\mathcal{G} = (V, E, W)$, in which vertices ($v \in V$) represent languages within the translation process, while an edge ($e \in E$) signifies a conditional transition from the current vertex to the next vertex. A path starts from the source, passes through multiple auxiliary languages, and ends with the target. A weight ($w \in W$) assigned to each edge represents the conditional probability of transitioning from the current vertex (i.e. language) to the next vertex, given the all previous vertices in the path.

We construct the meta-graph with five steps as the left side of Fig. 1: (1) **Vertex Initiation.** Create two vertices to represent the source language and target language respectively; (2) **Edge Establishing.** Connect the source vertex to m different auxiliary vertices. Each auxiliary vertex represents a unique auxiliary language from a set of m available options and is assigned a unique probability computed in the next paragraph. An edge shows a directed connection from the existing vertex to the new-connected vertex; (3) **Path Growth.** Further extend directed connections of each auxiliary vertex above to the target vertex or all of the other auxiliary vertices, which are not previously connected in the preceding path. (4) **Path Completion.** A path contains contiguous connections between vertices and is complete if it reaches the target vertex, otherwise continue extending by step (3); (5)

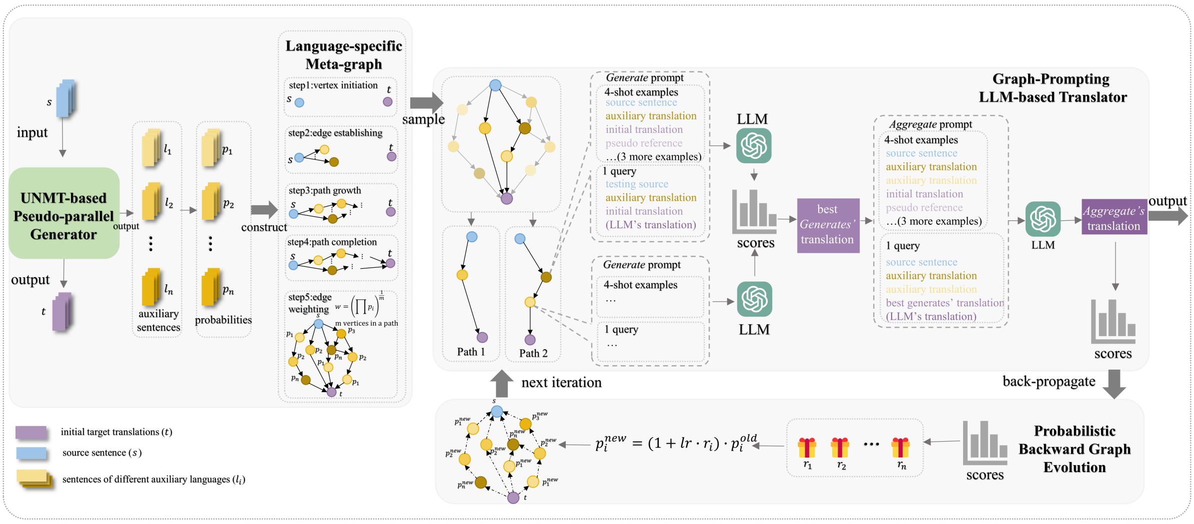

Figure 1: The architecture of POMP. Start from a source language dataset. (1) The UNMT-based pseudo-parallel generator translates a source dataset to the target as pseudo-parallel inputs in prompts and to auxiliary languages to calculate language similarities. (2) Language-specific meta-graph is constructed in 5 steps based on the language similarities. (3) Graph-prompting LLM-based translator samples translation paths from the meta-graph to organize auxiliary languages in prompts of *Generate* and *Aggregate* operations for LLMs. (4) Probabilities backward graph evolution derives rewards for each auxiliary language from the evaluation scores of translation outputs and updates probabilities in the meta-graph, which serves for the next iteration.

Edge Weighting. Assign a weight for each edge with the geometric average of all probabilities of previous vertices in the path.

In the meta-graph constructed above, we obtain the unique probability of an auxiliary language considering the language similarity between it and the source, since the probability is designed to guide the organization of auxiliary language in constructing prompts for LLMs. The operations are as follows: (1) We first encode n pseudo-parallel sentences of source-auxiliary pairs $(s_i, a_i), (i \in n)$ generated in §3.2 to get vector pairs $\mathbf{v}_{s_i}, \mathbf{v}_{a_i} = Encoder(s_i, a_i)$; (2) To measure a language similarity between the source and an auxiliary language, we average a cosine similarity ($\overline{\cos}$) of all vector pairs; (3) Finally, we scale the similarity to the probability of the auxiliary language as $p = \exp\left(-1 + \overline{\cos}\right)$.

Above all, we construct a language-specific meta-graph that carries relations among the source, target, and auxiliary languages which sample to generate multiple translation paths to prompt LLMs in §3.4.

3.4 Graph-prompting LLM-based Translator

We explore diverse improved translations by sampling multiple translation paths from the meta-graph in §3.3 to prompt an LLM as a translator for LRLs. Specifically, we employ two operations: *Generate* and *Aggregate*. First, we independently sample multiple paths from the language-specific meta-graph. In *Generate*, we construct prompts with the individual auxiliary language at a vertex of the path. In *Aggregate*, we combine all the auxiliary languages in a path to construct prompts. We prompt LLMs to generate multiple improved LRL translations respectively. See Fig. 4 as a prompt example.

3.4.1 Sample

To explore diverse improved LRL translations, we sample multiple translation paths from the meta-graph in §3.3 according to the probabilities of the paths. We obtain the probabilities of the paths by calculating the geometric average of the probabilities p_j of m auxiliary language in the path as $p_{[1,m]} = \left(\prod_{j=1}^{m} p_j\right)^{\frac{1}{m}}$. These sampled paths provide diverse auxiliary languages and their combination to construct prompts in operation *Generate* and *Aggregate*.

3.4.2 Generate

To make full use of auxiliary language and explore LLMs' ability, we execute a *Generate* operation for each vertex in a sampled path. As shown in Fig. 4, following ICL methods, we construct the prompt with 4-shot examples and 1 query. Each example consists of 4 parts: a source sentence, its pseudo-parallel auxiliary sentence at a vertex, its translation output from §3.2, and its pseudo reference from Google Translate as a refined translation. The query contains an input source sentence and its other parts except for the refined translation as the example. The LLM is prompted to generate a refined translation for the input.

We perform the *Generate* operation once for each of the m auxiliary language vertices on an input source sentence, yielding m translations. For an unsupervised evaluation without ground truth, we need to obtain pseudo ground truth: sequentially select each of the m translations as a pseudo ground truth, and then we evaluate the remaining translations considering the pseudo ground truth to get $m-1$ scores. The average of the $m-1$ scores stands for an unsupervised evaluation result (*gen-score*, e_i) of the selected (i-th) translation. In training and testing, we evaluate each output of *Generate* and feed the best output to *Aggregate* (§3.4.3) as its input.

3.4.3 Aggregate

To utilize multiple auxiliary languages and merge the useful information to feed into LLMs, we execute a *Aggregate* operation, in which we aggregate all auxiliary languages of the entire sampled path to prompt LLMs. Similar to *Generate*, the *Aggregate* operation applies the ICL paradigm with a 4-shot example and 1 query format as the illustrated example in Fig. 4. However, *Aggregate* differs from *Generate* as (1) it aggregates pseudo-parallel sentences of all auxiliary languages along the path into prompts, whereas *Generate* uses sentences of a single auxiliary language at each vertex; (2) the query in *Aggregate* includes the improved translation from the *Generate* instead of the original translation.

While *Generate* operates totally m times to produce m translations, *Aggregate* is conducted once, aggregating all auxiliary languages in the path to prompt LLMs, yielding one translation. For an unsupervised evaluation without ground truth, we regard the *Aggregate*'s translation as the pseudo-ground truth. We calculate the average score of evaluating the *Generate*'s m translations considering this pseudo ground truth as the unsupervised evaluation score (*agg-score*, E). In training, the evaluation score of *Aggregate*'s output carries back-propagated rewards to update probabilities of involved auxiliary languages in §3.5. In testing, we select the highest scoring one among the m *Generate*'s outputs and one *Aggregate* output as the final output.

3.5 Probabilistic Backward Graph Evolution

To make the meta-graph better serve for sampling translation paths to prompt LLMs (§3.4), we back-propagate evaluation scores in *Aggregate* operations to update probabilities of auxiliary languages in the meta-graph (§3.3), which consists of three steps: (1) quantify the contributions (d_i) of all individual auxiliary languages made for the translation, (2) normalize the contributions (d_i) to obtain rewards (r_i), (3) back-propagate the rewards (r_i) to update the probabilities (p_i^{new}) of auxiliary languages in the meta-graph. Details are as follows.

Quantify contributions. As all auxiliary languages in a sampled translation path contribute to an *Aggregate* operation's output, we quantify each auxiliary language i's contribution d_i to the translation output. First, we assume all m auxiliary languages used in the translation path equally contribute to the output translated via this translation path. As the translated output is measured by *agg-score* E in §3.4.3, we equally assign E to contributions as $E = \sum_i^m d_i$. Second, to reflect each language's contribution, we use *gen-score* e_i in §3.4.2 for auxiliary language i. Introducing e_i, we obtain the translation score without considering each language $E - e_i$, and the corresponding contribution without the language i is $\sum_i^{m-1} d_i$. Third, we apply the above calculations to all languages and obtain Eq. 1.

$$\begin{cases} E - e_1 & = d_2 + d_3 + \cdots + d_m \\ E - e_2 & = d_1 + d_3 + \cdots + d_m \\ & \vdots \\ E - e_m & = d_1 + d_2 + \cdots + d_{m-1} \end{cases} \tag{1}$$

Further, we obtain contribution d_i for each language i by combining m equations in Eq. 1 (See deductions in Appendix B).

$$d_i = E - \frac{1}{m-1} \left(\sum_{\substack{j=1 \\ j \neq i}}^{m} e_j \right) \tag{2}$$

Normalize contributions as rewards. We normalize the contribution d_i to obtain reward r_i, where r_i falls into a range of $[0, 1]$ so that the updated probability is constrained in a range of $[0, 1]$. We employ a central-symmetry Swish (cs-Swish) function (Ramachandran et al., 2017) to normalize d_i to r_i.

$$\text{cs-Swish}(x) = \begin{cases} -x \cdot \text{Sigmoid}(-x), & \text{if } x \leq 0; \\ x \cdot \text{Sigmoid}(x), & \text{if } x > 0. \end{cases} \quad (3)$$

We get r_i by combining Eq. 2 and Eq. 3 as:

$$\begin{aligned} r_i &= \text{cs-Swish}(d_i) \\ &= \begin{cases} -d_i \cdot \frac{1}{1+e^{d_i}}, & \text{if } d_i \leq 0; \\ d_i \cdot \frac{1}{1+e^{-d_i}}, & \text{if } d_i > 0. \end{cases} \end{aligned} \quad (4)$$

Update probabilities. For one training sample and its sampled translation path, we now obtain m respective rewards $r_i (i = 0, 1, \cdots, m)$ of the m auxiliary languages in the path. We back-propagate the r_i to update the probability (p_i^{new}) of the i-th auxiliary language with a learning rate (lr) and its previous probability (p_i^{old}) as

$$p_i^{\text{new}} = (1 + lr \cdot r_i) \cdot p_i^{\text{old}}. \quad (5)$$

As a result, one training iteration updates the probabilities of its involved auxiliary languages in the meta-graph, which continues to serve for the next iteration. When training converges, we use the translation paths sampled in the last iteration for testing.

4 Experiments

4.1 Experimental Settings

Languages. We select four LRL pairs, including Gujarati (Gu)→English (En), Kazakh (Kk)→En, Nepail (Ne)→En, and Sinhala (Si)→En to train their respective prompting graph with our approach and then to test. Following Chen et al. (2022), we utilize German (De), Spanish (Es), Finnish (Fi), Hindi (Hi), Russian (Ru), and Chinese (Zh) as the auxiliary languages, which are high-resource languages from different language families.

Datasets. To construct pseudo-parallel datasets for training, we collect datasets from the OPUS[2] (Tiedemann, 2012, 2016). Specifically, the datasets are from WMT (Gu, Kk) and CCAligned (Ne, Si). Then we randomly sample 1000 sentences for each dataset and translate the source side with the UNMT model in §3.2. The testing data is from newstest[3] (Gu, Kk) and Flores-200 Testset[4] (Team et al., 2022) (Ne, Si).

Evaluation Metrics. Recent work has shown that n-gram metrics like BLEU (Papineni et al., 2002) are sub-optimal for evaluating high-quality translations (Kocmi et al., 2021; Freitag et al., 2021). As recommended in Freitag et al. (2022), neural network-based metrics demonstrate a high correlation with human evaluation. Therefore, we adopt COMET[5] (Rei et al., 2020), xCOMET[5] (Guerreiro et al., 2023), and BLEURT[6] (Sellam et al., 2020) as our evaluation metric. Specifically, we use BLEURT-20[7] model in Pytorch implementation[8] for BLEURT metric. As for COMET and xCOMET, we use the wmt22-comet-da model[9] and XCOMET-XL model.[10]

Baselines. We compare with three non-LLM baselines: (1) `CRISS` (Tran et al., 2020), initialized with mBART and fine-tuned on 180 kinds of translation pairs from CCMatrix dataset; (2) `m2m-100` (Fan et al., 2020), a supervised multilingual NMT model trained with 7.5B parallel sentences from CCMatrix and CCAligned datasets; (3) `SixT+` (Chen et al., 2022), initialized with XLM-R-large (Goyal et al., 2021), learns on high-resource auxiliary language and inference on LRLs in an unsupervised way. We compare with three LLM baselines: (1) `ChatGPT-QS` (Hendy et al., 2023), investigating prompt learning on ChatGPT (OpenAI, 2022) on LRL setting; (2) `ChatGPT-ICL` (Zhu et al., 2023), applying in-context learning to achieve the translation of ChatGPT; (3) `ChatGPT-trans`, employing ICL in the prompt with 4-shot examples consisting of 4 source sentences and their pseudo-parallel pairs to ask ChatGPT to translate the testing set (See its prompt text in Fig. 5).

See the implementation details in Appendix A.

4.2 Overall performance

Tab. 1 shows the results of all comparing methods. POMP outperforms all the baselines on all metrics. In LRL translations of non-LLM baselines,

[2] opus.nlpl.eu

[3] data.statmt.org/wmt19/translation-task

[4] https://tinyurl.com/flores200dataset

[5] https://github.com/Unbabel/COMET/

[6] https://github.com/google-research/bleurt

[7] https://github.com/google-research/bleurt/blob/master/checkpoints.md

[8] https://github.com/lucadiliello/bleurt-pytorch

[9] https://huggingface.co/Unbabel/wmt22-comet-da

[10] https://huggingface.co/Unbabel/XCOMET-XL

Type	Model	Gu→En			Kk→En			Ne→En			Si→En		
		COMET	xCOMET	BLEURT	COMET	xCOMET	BLEURT	COMET	xCOMET	BLEURT	COMET	xCOMET	BLEURT
non-LLM	CRISS	79.88	65.25	62.42	72.80	52.34	54.86	83.54	73.33	63.99	80.66	62.01	61.22
	m2m-100	36.56	20.82	26.60	35.12	25.57	25.04	70.05	44.65	50.99	81.48	66.31	63.60
	SixT+	86.68	86.92	65.78	84.17	84.56	65.59	88.98	87.76	68.79	85.49	82.90	63.00
LLM	ChatGPT-QS	84.28	79.24	70.93	79.51	71.98	65.20	87.59	85.03	71.55	67.47	32.89	42.33
	ChatGPT-ICL	87.49	87.22	74.36	81.11	75.75	67.43	88.31	86.50	72.57	70.25	36.03	44.75
	ChatGPT-trans	85.99	87.06	71.87	78.45	76.70	63.81	86.16	84.12	68.96	59.63	30.42	34.07
	POMP	**88.55**	**91.52**	**75.22**	**84.77**	**88.10**	**71.87**	**89.66**	**91.43**	**74.88**	**86.28**	**86.64**	**70.21**

Table 1: Results of all methods on COMET, xCOMET, and BLEURT. The best results are in bold.

Model	Gu→En			Kk→En			Ne→En			Si→En		
	COMET	xCOMET	BLEURT	COMET	xCOMET	BLEURT	COMET	xCOMET	BLEURT	COMET	xCOMET	BLEURT
POMP	**88.55**	**91.52**	**75.22**	**84.77**	**88.10**	**71.87**	**89.66**	91.43	**74.88**	**86.28**	**86.64**	**70.21**
w/o auxiliary	87.78	89.46	74.00	84.62	83.33	71.43	89.58	91.43	74.74	85.61	83.04	69.47
w/o *Generate*	83.92	89.65	73.73	82.88	87.82	71.24	85.54	90.48	74.17	83.23	86.38	70.09
w/o *Aggregate*	88.44	90.03	74.88	84.40	86.34	71.42	89.62	91.39	74.47	85.70	81.18	69.30
w/o updating	87.99	88.97	74.35	84.16	85.90	70.94	89.64	**91.66**	74.75	85.38	79.67	68.91
w/o scoring	88.05	89.44	74.35	83.81	85.12	70.94	89.49	91.17	74.75	85.11	78.07	68.91
w/o meta-graph	84.53	91.20	74.46	82.97	87.54	71.30	85.84	90.76	74.39	82.37	85.22	68.79

Table 2: Ablation study. w/o indicates our full model without the specific component.

POMP outperforms unsupervised CRISS, supervised m2m-100, and state-of-the-art SixT+, which indicates the effectiveness of our approach. In LRL translations of LLM baselines, we compare the POMP with the other three LLM-based baselines: ChatGPT-QS, ChatGPT-ICL, and ChatGPT-trans. They all fall behind POMP, which verifies the effectiveness of our strategy of constructing prompts. Notably, for Ne→En translations, all LLM-based baselines underperform non-LLM baselines, indicating that LLMs hardly do well in LRL translations. POMP improves Ne→En translations for LLMs to outperform non-LLM baselines, demonstrating our approach's effectiveness. Additionally, we conduct experiments on the GPT-4 and show results at Tab. 5 in Appendix C, which are also better than the initial UNMT system. POMP has similar performance on different LLMs, indicating its generality.

4.3 Ablation Study

Tab. 2 presents ablation studies on our model. Our full model outperforms in all metrics across four LRLs, except for the variant w/o updating on the xCOMET metric of Ne. The variant w/o auxiliary, removing all auxiliary languages in prompts, underperforms POMP, which shows helpful assistance of auxiliary languages. W/o *Generate* excludes *Generate* operations from POMP, focusing solely on *Aggregate* to combine all auxiliary languages of a translation path into prompts. Conversely, w/o *Aggregate* excludes *Aggregate* from POMP, using a single auxiliary language in a prompt. Both of them fall behind POMP, indicating that the two operations play vital roles in POMP. In w/o updating, we fix the probabilities of selecting auxiliary languages as their initial ones calculated by language similarities in §3.3, of which the results show the validity of the graph evolution strategy in §3.5. While the Ne→En translation in w/o updating excels POMP, indicating that Ne's language similarities with auxiliary languages provide solid probabilities for organizing them in prompts. W/o scoring represents the strategy in which we randomly choose the final translation output among translations of *Generate* and *Aggregate*. Its poor performances verify the success of our unsupervised evaluation methods in §3.4. W/o meta-graph remove the constructed meta-graph (§3.3) and randomly organize the auxiliary languages in prompts for LLMs. Results show that the translation paths sampling from the meta-graph offer practical improvements to prompt LLMs in LRL translations.

4.4 Analysis of the Linguistically Biased Translation Errors

We attempt to quantify the linguistically biased translation errors of POMP versus some straightforward usages. Specifically, we capture the linguistic characteristics (i.e. output token distributions) on (1) the LLM's results using a single auxiliary language (1-auxiliary), (2) our full model's results, and (3) the ground truth. Then, we measure POMP's linguistically biased errors by evaluating

	Gu→En	Kk→En	Ne→En	Si→En
1-auxiliary	0.7365	0.7254	0.7222	0.7198
POMP	**0.2407**	**0.2600**	**0.2109**	**0.2592**

Table 3: JSD of 1-auxiliary's token distributions and POMP's token distributions against the ground truth across the four LRLs.

the distribution gap between the ground truth and POMP's results; we measure 1-auxiliary's linguistically biased errors by evaluating the distribution gap between the ground truth and 1-auxiliary's results. We use Jensen-Shannon divergence (JSD) to quantify the distribution gap. In Tab. 3, 1-auxiliary's JSD is higher, showing that a single auxiliary language suffers from biases from this language and is likely to cause translation errors in LLM's output. Conversely, POMP achieves much smaller distribution gaps across 4 LRLs, indicating fewer translation errors in POMP's results.

We also visualize the gaps in Fig. 2 and Fig. 3 Appendix D. We apply kernel density estimation (KDE) to estimate the discrete token frequencies as continuous curves, which sensitively reflect slight variations in the shapes of distributions. We observe that POMP's distribution matches well with the reference across 4 LRLs, while 1-auxiliary's distribution falls far behind.

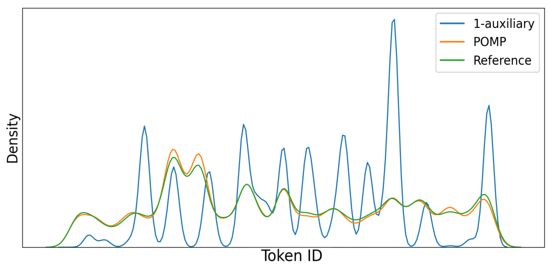

Figure 2: The visualization of token distributions of 1-auxiliary (blue), POMP (yellow), and the reference (green). The horizontal axis represents the token IDs appearing in involved sentences, and the vertical axis represents the kernel density of token frequencies.

4.5 Analysis of the Prompting Graph

To explore the fitness of POMP's prompting graphs, we analyze prompting graphs' structures by measuring degrees of vertices and lengths of translation paths. As a vertex represents a language, we measure the degrees of 8 kinds of vertices (source, target, 6 auxiliary languages). The degrees mean the connections of vertices, which measure the usages of auxiliary languages. The length of a translation path represents the number of edges in the path. The average degree of total vertices is 3.27, indicating that vertices in prompting graphs connect well with each other. The lengths of translation paths range from $[2, 7]$, corresponding to the involvement of one to six auxiliary languages. This range suggests the effective organization of auxiliary languages for LRL translations, facilitating efficient translation pathways. The average length of total translation paths is 3.42, indicating that POMP utilizes 2 or 3 auxiliary languages to prompt LLMs. The results in Tab. 4 show that graphs of each testing language in POMP fit well for LLMs' prompts.

4.6 Case Study

We show the cases of translation outputs of POMP versus baselines in Fig. 9 and Fig. 10. POMP generates fluent and accurate translations in all cases. Non-LLM baselines like CRISS and m2m-100 are likely to mistranslate words and repeat meaningless words, while SixT+ achieves better but less accuracy than POMP. LLMs-based baselines tend to generate facts unrelated to the given source. As a comparison, cases show that POMP's outputs with more precise words and fewer repetitions are consistent with references.

5 Conclusion

In summary, we propose POMP, an unsupervised method that mitigates linguistically biased translation errors in UNMT on LRLs. This approach involves constructing a language-specific meta-graph, from which we sample multiple translation paths with organized auxiliary languages to prompt LLMs as LRL translators. We promote the evolution of the meta-graph by back-propagating evaluation scores to update probabilities of involved auxiliary languages in the graph. We use three metrics for evaluations in testing. Our experiment results demonstrate the effectiveness of our approach, which achieves SOTA performances on four LRLs.

6 Limitations

In our work, there are several limitations. (1) We use GPT-3.5 in our approach rather than GPT-4, and GPT-3.5 is not the most advanced GPT API currently available, which seems to limit the performance of our model. The main reason is that LLM-based baselines use GPT-3.5, we follow them

	Gu→En	Kk→En	Ne→En	Si→En	Avg.
src	3.24	8.29	7.70	4.38	5.90
tgt	3.24	8.29	7.70	4.38	5.90
De	0.86	1.62	1.67	1.23	1.35
Es	1.48	3.20	4.28	2.19	2.79
Fi	1.57	4.88	4.05	1.91	3.10
Hi	1.10	4.61	1.83	1.24	2.20
Ru	1.34	1.97	3.43	1.84	2.15
Zh	1.46	3.95	3.47	2.13	2.75
length	3.41	3.44	3.43	3.40	3.42

Table 4: Results of graphs of each testing language in POMP fit for prompting LLMs. Languages represent types of vertices and length represents the length of average translation paths.

for a fair comparison. Meanwhile, the cost of utilizing GPT-4 is significantly higher than that of GPT-3.5. According to the official website, the fee for GPT 3.5 is 0.002$/1k Token, and the fee for GPT 4 is 0.06$/1k Token. In our experiments, the API fee of GPT-3.5 costs about 130$ in total. If the GPT 4 is used, the same number of tokens will cost about 3900$ in total. Hence, employing GPT-3.5 not only results in a substantial reduction in costs but also yields a largely comparable generation effect. Currently, we are utilizing GPT-3.5 for the validation of our methods, with plans to employ GPT-4 for effect validation in subsequent phases. (2) POMP is limited for translation tasks that require precise words (evaluated by BLEU-like score). Since LLMs are pre-trained and fine-tuned with large amounts of data in various domains, they prefer diverse generations. We encourage future works with a constrained dictionary and precise instructions in prompts to achieve accurate translations.

7 Ethical Considerations

We take ethical considerations very seriously, and strictly adhere to the ACL Ethics Policy. (1) LLM training predominantly relies on data from high-resource languages, potentially leading to biases in low-resource languages (LRL) translation, exacerbating linguistic and cultural biases. (2) LLMs may lack exposure to specific customs, beliefs, or values prevalent in LRL, resulting in translation inaccuracies, misunderstandings, and potential offense. To mitigate these ethical challenges, we plan to implement restrictions within the prompt templates to discourage the use of discriminatory or biased language. Moreover, we plan to compile a sensitivity lexicon to identify sensitive words, enabling us to either avoid translating them or to find alternative translations. Thus, we believe that the ethical issues raised in this research can be handled with some carefully designed strategies and thus the usage of our model would not cause serious ethical problems.

Acknowledgements

This work is supported by the following foundations: the National Natural Science Foundation of China under Grant No.62025208, the Young Elite Scientist Sponsorship Program by CAST (2023QNRC001) under Grant No.YESS20230367, the National Natural Science Foundation of China under Grant No.62306330, No.62106275, the Xiangjiang Laboratory Foundation under Grant No.22XJ01012, and the Natural Science Foundation of Hunan Province under Grant No.WDZC20235250103.

References

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2023. In-context examples selection for machine translation. In *Findings of ACL 2023*, pages 8857–8873, Toronto, Canada. Association for Computational Linguistics.

Duarte Alves, Nuno Guerreiro, João Alves, José Pombal, Ricardo Rei, José de Souza, Pierre Colombo, and Andre Martins. 2023. Steering large language models for machine translation with finetuning and in-context learning. In *Findings of EMNLP 2023*, pages 11127–11148, Singapore. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *ICLR 2018*.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michał Podstawski, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2023. Graph of Thoughts: Solving Elaborate Problems with Large Language Models.

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 *NeurIPS*

2020, volume 33, pages 1877–1901. Curran Associates, Inc.

Shweta Singh Chauhan, Shefali Saxena, and Philemon Daniel. 2022. Improved unsupervised neural machine translation with semantically weighted back translation for morphologically rich and low resource languages. *Neural Processing Letters*, 54:1707 – 1726.

Guanhua Chen, Shuming Ma, Yun Chen, Dongdong Zhang, Jia Pan, Wenping Wang, and Furu Wei. 2022. Towards making the most of cross-lingual transfer for zero-shot neural machine translation. In *ACL 2022*, pages 142–157, Dublin, Ireland. Association for Computational Linguistics.

Peng-Jen Chen, Ann Lee, Changhan Wang, Naman Goyal, Angela Fan, Mary Williamson, and Jiatao Gu. 2020. Facebook AI's WMT20 news translation task submission. In *WMT 2020*, pages 113–125, Online. Association for Computational Linguistics.

Alexandra Chronopoulou, Dario Stojanovski, and Alexander Fraser. 2021. Improving the lexical ability of pretrained language models for unsupervised neural machine translation. In *NAACL 2021*, pages 173–180, Online. Association for Computational Linguistics.

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 *ACL 2020*, pages 8440–8451, Online. Association for Computational Linguistics.

Anna Currey and Kenneth Heafield. 2019. Zero-resource neural machine translation with monolingual pivot data. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 99–107, Hong Kong. Association for Computational Linguistics.

Raj Dabre, Tetsuji Nakagawa, and Hideto Kazawa. 2017. An empirical study of language relatedness for transfer learning in neural machine translation. In *Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation*, pages 282–286. The National University (Phillippines).

Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. 2023. Active prompting with chain-of-thought for large language models.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning.

Lukas Edman, Antonio Toral, and Gertjan van Noord. 2020. Low-resource unsupervised NMT: Diagnosing the problem and providing a linguistically motivated solution. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 81–90, Lisboa, Portugal. European Association for Machine Translation.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *EMNLP 2018*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *arXiv preprint*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(1).

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In *WMT 2022*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Xavier Garcia, Yamini Bansal, Colin Cherry, George Foster, Maxim Krikun, Melvin Johnson, and Orhan Firat. 2023. The unreasonable effectiveness of few-shot learning for machine translation. In *ICML 2023*, ICML'23. JMLR.org.

Naman Goyal, Jingfei Du, Myle Ott, Giri Anantharaman, and Alexis Conneau. 2021. Larger-scale transformers for multilingual masked language modeling. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 29–33, Online. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of ACL 2022*, 10:522–538.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2023. xcomet: Transparent machine translation evaluation through fine-grained error detection.

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are gpt models at machine translation? a comprehensive evaluation.

Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2023. Is chatgpt a good translator? yes with gpt-4 as the engine.

Yunsu Kim, Petre Petrov, Pavel Petrushkov, Shahram Khadivi, and Hermann Ney. 2019. Pivot-based transfer learning for neural machine translation between non-English languages. In *EMNLP-IJCNLP 2019*, pages 866–876, Hong Kong, China. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *WMT 2021*, pages 478–494, Online. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *ICLR 2018*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. In *EMNLP 2018*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.

Yichong Leng, Xu Tan, Tao Qin, Xiang-Yang Li, and Tie-Yan Liu. 2019. Unsupervised pivot translation for distant languages. In *ACL 2019*, pages 175–183, Florence, Italy. Association for Computational Linguistics.

Jiahuan Li, Hao Zhou, Shujian Huang, Shanbo Cheng, and Jiajun Chen. 2023. Eliciting the translation ability of large language models via multilingual finetuning with translation instructions.

Zhaocong Li, Xuebo Liu, Derek F. Wong, Lidia S. Chao, and Min Zhang. 2022. ConsistTL: Modeling consistency in transfer learning for low-resource neural machine translation. In *EMNLP 2022*, pages 8383–8394, Abu Dhabi, United Arab Emirates. Association for 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 *EMNLP 2022*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chao-Hong Liu, Catarina Cruz Silva, Longyue Wang, and Andy Way. 2019. Pivot machine translation using chinese as pivot language. In *Machine Translation*, pages 74–85, Singapore. Springer Singapore.

OpenAI. 2022. https://openai.com/blog/chatgpt.

OpenAI. 2023. Gpt-4 technical report.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL 2002*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2023. Towards making the most of ChatGPT for machine translation. In *Findings of EMNLP 2023*, pages 5622–5633, Singapore. Association for Computational Linguistics.

Prajit Ramachandran, Barret Zoph, and Quoc V. Le. 2017. Swish: a self-gated activation function. *arXiv: Neural and Evolutionary Computing*.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *EMNLP 2020*, pages 2685–2702, Online. Association for Computational Linguistics.

Felipe Sánchez-Martínez, Víctor M. Sánchez-Cartagena, Juan Antonio Pérez-Ortiz, Mikel L. Forcada, Miquel Esplà-Gomis, Andrew Secker, Susie Coleman, and Julie Wall. 2020. An English-Swahili parallel corpus and its use for neural machine translation in the news domain. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 299–308, Lisboa, Portugal. European Association for Machine Translation.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021a. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *EACL 2021*, pages 1351–1361, Online. Association for Computational Linguistics.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021b. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *ACL 2021*, pages 6490–6500, Online. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *ACL 2020*, pages 7881–7892, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Kashun Shum, Shizhe Diao, and Tong Zhang. 2023. Automatic prompt augmentation and selection with chain-of-thought from labeled data. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12113–12139, Singapore. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218. Citeseer.

Jörg Tiedemann. 2016. OPUS – parallel corpora for everyone. In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation: Projects/Products*, Riga, Latvia. Baltic Journal of Modern Computing.

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

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. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Chau Tran, Yuqing Tang, Xian Li, and Jiatao Gu. 2020. Cross-lingual retrieval for iterative self-supervised training. In *Advances in Neural Information Processing Systems*, volume 33, pages 2207–2219. Curran Associates, Inc.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2023. Prompting PaLM for translation: Assessing strategies and performance. In *ACL 2023*, pages 15406–15427, Toronto, Canada. Association for Computational Linguistics.

Zhihua Wen, Zhiliang Tian, Zexin Jian, Zhen Huang, Pei Ke, Yifu Gao, Minlie Huang, and Dongsheng Li. 2024. Perception of knowledge boundary for large language models through semi-open-ended question answering.

Wikipedia. 2024. https://en.wikipedia.org/wiki/google_translate.

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.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Multilingual machine translation with large language models: Empirical results and analysis.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *EMNLP 2016*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

A Implementation Details

We implement POMP based on an open-source translation framework Fairseq[11] and LLM framework GoT[12] (Besta et al., 2023). We perform POMP's training on 1 GeForce RTX 3090 GPU with 24GB memory and testing on 1 Tesla V100 GPU with 32GB memory.

We train the UNMT-based pseudo-parallel generator with 100k and 10k steps for the first and second stages. The batch size is adapted with 5000

[11] github.com/facebookresearch/fairseq
[12] github.com/spcl/graph-of-thoughts

Model	Gu→En			Kk→En			Ne→En			Si→En		
	COMET	xCOMET	BLEURT	COMET	xCOMET	BLEURT	COMET	xCOMET	BLEURT	COMET	xCOMET	BLEURT
SixT+	86.68	86.92	65.78	84.17	84.56	65.59	88.98	87.76	68.79	85.49	82.90	63.00
POMP	**88.55**	**91.52**	**75.22**	84.77	88.10	71.87	89.66	91.43	74.88	86.28	**86.64**	70.21
$\text{POMP}_{\text{gpt4}}$	88.30	91.03	74.85	**84.82**	**88.12**	**72.05**	**89.79**	**92.78**	**75.28**	**86.76**	86.01	**71.93**

Table 5: Results of POMP's performance on the GPT-4 ($\text{POMP}_{\text{gpt4}}$) and ChatGPT (POMP) on COMET, xCOMET, and BLEURT.

max tokens. The beam size is 5. We use the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. The first and second stages' learning rates are 5×10^{-4} and 1×10^{-4}. The warmup steps of the first and second stages are 4k and None. The probability of dropout is 0.1.

The dimension of vectors for calculating language similarities is 1024. We select "gpt-3.5-turbo-1106" with a maximum of 16k input tokens and use the identical prompt template shown in Fig. 4-Fig. 8 via the OpenAI API to generate translations of POMP and LLM-based baselines. The temperature and max tokens for LLMs' generation are 1.0 and 512.

We do not compare with Google Translate, which assembles many SOTA works and utilizes a great amount of data and resources, including human annotations, feature engineering, linguistic resources, and the latest LLMs (Wikipedia, 2024). Therefore, it is unfair to compare with Google Translate as baselines.

B Details for Solving Eq. 1

To directly derive a formula for d_i from the equations, we follow a mathematical process:

Given the system of equations:

$$\begin{cases} E - e_1 & = d_2 + d_3 + \cdots + d_m \\ E - e_2 & = d_1 + d_3 + \cdots + d_m \\ & \vdots \\ E - e_m & = d_1 + d_2 + \cdots + d_{m-1} \end{cases}$$

we want to calculate the expression for d_i.

1. **Sum all equations** to get a total that combines each $E - e_i$ term:

$$mE-(e_1+e_2+\cdots+e_m) = (m-1)(d_1+d_2+\cdots+d_m)$$

2. **Solve for the total sum of d_i's**:

$$\sum_{i=1}^{m} d_i = \frac{mE - \sum_{i=1}^{m} e_i}{m-1}$$

3. **Subtract the sum of all e_i's except e_i from** mE to isolate the contributions relevant to d_i:

$$\text{Total without } e_i = mE - \sum_{\substack{j=1 \\ j \neq i}}^{m} e_j$$

4. **Derive** d_i by recognizing it's the missing piece in each equation's total sum, balanced by $m-1$:

$$d_i = E - \frac{1}{m-1} \left(\sum_{\substack{j=1 \\ j \neq i}}^{m} e_j \right)$$

C Performance on the GPT-4

We conduct experiments on the GPT-4 ($\text{POMP}_{\text{gpt4}}$) and show results at Tab. 5. The additional results on GPT-4 are also better than the initial UNMT system. POMP has similar performance on different LLMs, indicating its generality.

D Visualizations of Distribution Gaps

We apply kernel density estimation (KDE) for the discrete token frequencies to visualize the distributions in continuous curves. The horizontal axis represents the token IDs that have appeared in involved sentences, and the vertical axis represents the kernel density. While KDE distributions of POMP perform well with the references, 1-auxiliary's distributions fall further behind than JDS values in 3 do. The reasons are as follows: (1) KDE can sensitively reflect slight variations in the shapes of distributions, which might not be as apparent in JSD calculations. (2) Distribution graphs reveal differences in the shape between distributions, such as the location and width of peaks. These shape differences might contribute less to JSD than to visual impressions. (3) If the distributions are multimodal (i.e., they have multiple peaks), distribution graphs might show significant visual differences even if JSD is relatively small. Overall, Fig. 3 shows the visualized significant effectiveness of our approach in mitigating linguistically biased translation errors.

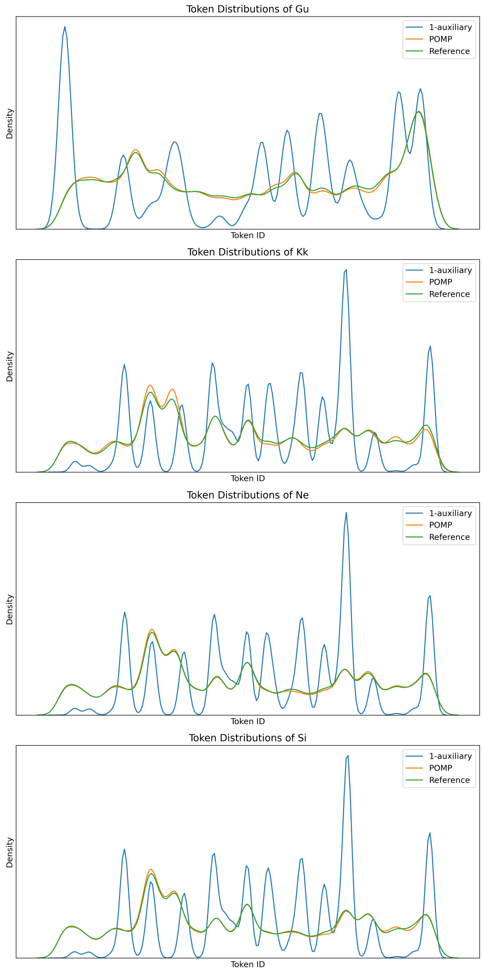

Figure 3: Visualizations of kernel density estimation (KDE) distribution gaps of tokens in different translation pairs.

type of prompt	components	context
prompt in Generate with Spanish	1 example	<Sinhala source>: පොලිසිය පවසන පරිදි, ඡායාරූප ශිල්පියා හැපුණු වාහනයේ රියදුරුට අපරාධ චෝදනා එල්ල වීමට ඉඩක් නැත. <Spanish translation>: Según las autoridades policiales, el conductor del vehículo que fue llevado por el fotógrafo no tiene ninguna posibilidad de ser acusado de delito. <English translation>: According to the police, the driver of the car where the photographer was kidnapped has no chance of being charged with a crime. <Refined translation>: According to police, the driver of the vehicle that hit the photographer is unlikely to face criminal charges.
	3 more examples	…
	1 query	<Sinhala source>: ඔවුහු සියලු දෙනා අනතුර සිදුවී තිබූ ස්ථානයේ ආපසු දිව ගියහ. <Spanish translation>: Todos volvieron de vuelta desde el lugar donde se produjo el accidente. <English translation>: They all returned from the location where the accident occurred. <Refined translation>:
prompt in Generate with Chinese	1 example	<Sinhala source>: පොලිසිය පවසන පරිදි, ඡායාරූප ශිල්පියා හැපුණු වාහනයේ රියදුරුට අපරාධ චෝදනා එල්ල වීමට ඉඩක් නැත. <Chinese translation>: 警方表示, 被摄像人乘坐的汽车司机没有可能被指控犯有罪行。 <English translation>: According to the police, the driver of the car where the photographer was kidnapped has no chance of being charged with a crime. <Refined translation>: According to police, the driver of the vehicle that hit the photographer is unlikely to face criminal charges.
	3 more examples	…
	1 query	<Sinhala source>: ඔවුහු සියලු දෙනා අනතුර සිදුවී තිබූ ස්ථානයේ ආපසු දිව ගියහ. <Chinese translation>: 他们全都从事故发生地跑了回去。 <English translation>: They all returned from the location where the accident occurred. <Refined translation>:
prompt in Aggregate with Spanish and Chinese	1 example	<Sinhala source>: පොලිසිය පවසන පරිදි, ඡායාරූප ශිල්පියා හැපුණු වාහනයේ රියදුරුට අපරාධ චෝදනා එල්ල වීමට ඉඩක් නැත. <Spanish translation>: Según las autoridades policiales, el conductor del vehículo que fue llevado por el fotógrafo no tiene ninguna posibilidad de ser acusado de delito. <Chinese translation>: 警方表示, 被摄像人乘坐的汽车司机没有可能被指控犯有罪行。 <English translation>: According to the police, the driver of the car where the photographer was kidnapped has no chance of being charged with a crime. <Refined translation>: According to police, the driver of the vehicle that hit the photographer is unlikely to face criminal charges.
	3 more examples	…
	1 query	<Sinhala source>: ඔවුහු සියලු දෙනා අනතුර සිදුවී තිබූ ස්ථානයේ ආපසු දිව ගියහ. <Spanish translation>: Todos volvieron de vuelta desde el lugar donde se produjo el accidente. <Chinese translation>: 他们全都从事故发生地跑了回去。 <English translation>: They all returned from the location where the accident occurred. <Refined translation>:

Figure 4: A sample of prompts in the translation path "Sinhala→Spanish→Chinese→English". In this sample. there are three types of prompts. The first two are the prompts in *Generate* at a vertex level, involving Spanish and Chinese pseudo-parallel sentences respectively. The last is the prompt in *Aggregate* at a path level, involving Spanish and Chinese pseudo-parallel sentences together.

type of prompt	components	context
prompt in ChatGPT-trans	1 example	<Sinhala source>: පොලිසිය පවසන පරිදි, ඡායාරූප ශිල්පියා හැපුණු වාහනයේ රියදුරුට අපරාධ චෝදනා එල්ල වීමට ඉඩක් නැත. <English translation>: According to the police, the driver of the car where the photographer was kidnapped has no chance of being charged with a crime.
	3 more examples	…
	1 query	<Sinhala source>: ඔවුහු සියලු දෙනා අනතුර සිදුවී තිබූ ස්ථානයේ ආපසු දිව ගියහ. <English translation>:

Figure 5: A sample of prompts in the baseline ChatGPT-trans. ChatGPT-trans utilizes a 4-shot ICL framework, in which an example consists of a Sinhala sentence and its target translation, and the LLM is prompted to generate the target translation from a given testing sentence in a query.

type of prompt	components	context
prompt in ChatGPT-QS	1 example	"පොලිසිය පවසන පරිදි, ඡායාරූප ශිල්පියා හැපුණු වාහනයේ රියදුරුට අපරාධ චෝදනා එල්ල වීමට ඉඩක් නැත." can be translated to "According to the police, the driver of the car where the photographer was kidnapped has no chance of being charged with a crime."
	3 more examples	…
	1 query	"ඔවුහු සියලු දෙනා අනතුර සිදුවී තිබූ ස්ථානයේ ආපසු දිව ගියහ." can be translated to

Figure 6: A sample of prompts in the ChatGPT-QS. An example in ChatGPT-QS includes a Sinhala sentence and a pseudo reference translation (e.g. from Google Translation) connected by the instruction "can be translated to", with a query showing the testing sentence.

type of prompt	components	context
prompt in ChatGPT-ICL	1 example	”පොලිසිය පවසන පරිදි, ඡායාරූප ශිල්පියා හැපුණු වාහනයේ රියදුරුට අපරාධ චෝදනා එල්ල වීමට ඉඩක් නැත.” = ”According to the police, the driver of the car where the photographer was kidnapped has no chance of being charged with a crime.”
	3 more examples	…
	1 query	”ඔවුහු සියලු දෙනා අනතුර සිදුවී තිබූ ස්ථානයෙන් ආපසු දිව ගියහ.” =

Figure 7: A sample of prompts in the ChatGPT-ICL. An example in ChatGPT-ICL includes a Sinhala sentence and a pseudo reference translation (e.g. from Google Translation) connected by "=", with a query showing the testing sentence.

type of prompt	components	context
prompt in ChatGPT-refine	1 example	<Sinhala source>: පොලිසිය පවසන පරිදි, ඡායාරූප ශිල්පියා හැපුණු වාහනයේ රියදුරුට අපරාධ චෝදනා එල්ල වීමට ඉඩක් නැත. <English translation>: According to the police, the driver of the car where the photographer was kidnapped has no chance of being charged with a crime. <Refined translation>: According to police, the driver of the vehicle that hit the photographer is unlikely to face criminal charges.
	3 more examples	…
	1 query	<Sinhala source>: ඔවුහු සියලු දෙනා අනතුර සිදුවී තිබූ ස්ථානයෙන් ආපසු දිව ගියහ. <English translation>: They all returned from the location where the accident occurred. <Refined translation>:

Figure 8: A sample of prompts in the ChatGPT-refine. An example in ChatGPT-refine includes the same Sinhala sentence, a translation from the NMT model, and a pseudo reference translation (e.g. from Google Translation), with a query showing the same testing sentence and its translation from the NMT model, prompting the LLM to refine this translation.

source	උණුසුම් චොකලට් බෙල්ජියම් සම්මතවලට අනුකූල වේ. පළතුරු යුෂ මිලෙන් වැඩි නමුත් අනර්ඝයි.
reference	The hot chocolate is up to Belgian standards. Fruit juices are pricey but excellent.
POMP	The hot chocolate meets the Belgian standards. The fruit juice is more expensive, but it is excellent.
CRISS	The hot chocolates comply with the standards of Belgium. (...)
m2m-100	Hot Chocolate complies with Belgian standards. The fruit is more expensive than juice, but it is uncomfortable.
SixT+	The hot chocolate conforms to the Belgian norms. The fruit juice is more expensive but exquisite.
ChatGPT-QS	The botanical gardens of Belém are connected to a large collection of plants. The first paragraph does not seem to be related to the rest of the text. Could you please provide more context or clarify the connection between these paragraphs?
ChatGPT-ICL	Eunice Chonglat of Belgium is related to the family of anthropologists. The first name is not clear and is ambiguous.
ChatGPT-trans	The chocolate industry is a major contributor to the economy. The exports generate a significant amount of revenue.

Figure 9: An example of generated translations of different baselines and our approach in Si→En. We highlight obvious translation errors in sentences in red. Note that the Google translation of the source is "*Hot chocolate conforms to Belgian standards. Fruit juice is more expensive but precious.*".

source	તુવેર લગભગ ચારેક મહનિા સુધી સારી રહી શકે છે ત્યાર પછી તેમાં ડંખ પડવા માંડે છે, સડવા માંડે છે.
reference	Toor can remain good for almost four months after which it starts to smell and begins to rot.
POMP	The tulip can stay fresh for about four months, after which it starts to wilt and rot.
CRISS	When it starts to grow, it starts to bloom, it starts to sting, it starts to thrive, it starts to sting, it starts to bloom, it starts to thrive, it starts to sting, it starts to sting, it starts to sting, it starts to sting, it starts to sting, it starts to sting, it starts to sting.
m2m-100	باء بضم الأولى " رقا مفْ ال ل كلّ ما ب والشَّيْ " أم " رقا مفْ ال ل كلّ ما والشَّيْبُ " . بفتحها والثانية ، الشيب
SixT+	The tulip can stay good for about four and a half months. After that, it starts to stain, rot.
ChatGPT-QS	The lentils can be stored for almost four months and then they need to be sieved.
ChatGPT-ICL	Toover can last for about four months, then it might get infested with worms, get contaminated.
ChatGPT-trans	You can keep the pipe well for about four months, then it needs to be cleaned and flushed.

Figure 10: An example of generated translations of different baselines and our approach in Gu→En. We highlight obvious translation errors in sentences in red. Note that the Google translation of the source is "*The tuber can stay good for about four months after which it starts to bite, rot.*" and the Google translation of the m2m-100's output is "*And the gray hair is not caused by God to be separated, "um" and by the gray hair is not caused by God to be separated. The first is with the ba' of al-shayb enclosed, and the second is with its opening.*".