Low-resource Neural Machine Translation with Large Language Models: A Continuous Self-Improving System

Anonymous ACL submission

Abstract

Machine translation systems often struggle with maintaining quality in low-resource scenarios, due to the lack of sufficient parallel data. We present a novel learning framework that continuously (potentially life-long) improves Large Language Model (LLM)'s performance for low-resource language machine translation through self-optimization. Our system comprises three key components: an Instruction Optimizer that dynamically refines translation prompts based on failure cases, a Demonstration Manager that intelligently selects relevant examples for in-context learning, and a Quality Estimator using multiple metrics that evaluates and arranges translations for the Instruction Optimizer and the Demonstration Manager. The resulting system, called DAIL-translation, boosts the performance in low-resource machine translation of moderate-sized LLMs (∼7B), larger-scale LLMs (∼70B) and OpenAI model series, with only 1k monolingual English sentences as a starting point.

1 Introduction

LLMs have demonstrated significant potential in the field of natural language processing (Yang et al., 2024b; OpenAI, 2023; Dubey et al., 2024). Some studies (Enis and Hopkins, 2024; Robinson et al., 2023; Zhu et al., 2024a) have shown that these models perform well in neural machine translation (NMT) tasks for high-resource languages but struggle with low-resource languages. Although most languages spoken worldwide today are low-resource languages, many languages within this category receive limited attention and resources (Joulin et al., 2017; Costa-jussà et al., 2022). Additionally, the data for low-resource languages is often scarce and difficult to find online. Therefore, machine translation for low-resource languages continues to be a challenging problem.

Effective methods for enhancing LLM capabilities primarily include: (1) Post-training methods such as Supervised Fine-Tuning (SFT), Direct Preference Optimization (DPO) (Rafailov et al., 2023) have demonstrated potential in improving model performance. However, as indicated by (Vieira et al., 2024), SFT can negatively impact model performance in machine translation tasks when training data is limited. While Contrastive Preference Optimization (CPO) (Xu et al., 2024) achieves great results in machine translation, its effectiveness is mainly verified on high-resource languages. (2) Prompt Engineering addresses the prompt-sensitive nature of LLMs, which significantly affects interaction outcomes. Nevertheless, automated prompt engineering methods (Yang et al., 2024c; Wang et al., 2024) often require performance of historical prompts as feedback signals, necessitating frequent and costly calls to LLMs. (3) In-Context Learning (ICL), by integrating examples into prompts, can enhance a model's ability to understand semantics and formats. However, according to (Court and Elsner, 2024), LLMs exhibit poor retrieval performance for low-resource languages, particularly when translating from low-resource languages to English. This issue arises due to difficulties in obtaining accurate text embeddings due to insufficient training data, leading to failures in similarity-based retrieval.

To tackle the challenges and better apply these effective methods to low-resource language translation, we have designed the DAIL-translation system. Our system is structured around two databases and three key components. One database stores accurate translations, denoted as $\{(g_q, g_t)\}$, facilitating ICL sampling; whereas the other retains potentially wrong translations, denoted as $\{(b_q, b_t)\}$, which aids in prompt optimization. The **Instruction Optimizer** dynamically refines translation prompts by analyzing stored failure cases, thus reducing the dependency on costly, frequent interactions with LLMs for automated prompt engineering. Our research also indicates that the length ratio

between input and output in high-quality translation pairs aligns with a language-specific Gaussian distribution. Consequently, when selecting ICL examples, the **Demonstration Manager** draws from this length ratio distribution to enhance format comprehension, complementing traditional similarity-based approaches for better semantic capturing. After translation, the **Quality Estimator** evaluates the output quality, deciding which database the translation should populate, thereby expanding the ICL search space for the Demonstration Manager or providing more bad cases for the Instruction Optimizer. Beyond length ratio, we have also identified that the perplexity ratio between input and output should be constrained within a certain range. Therefore, we employ both length ratio and perplexity as indicators for selection. Through these interconnected components, DAIL-translation demonstrates a robust, self-improving mechanism that significantly boosts the performance of LLMs across different scales, including moderate-sized models (∼7B parameters), larger models (∼70B parameters), and those within the OpenAI series.

Our contributions are:

- We propose DAIL-translation, a continuous self-improving system to enhance the translation ability of LLMs in low-resource languages without training and using only monolingual English data.
- With the help of past translations, we build an Instruction Optimizer that dynamically refines prompts for better translation quality.
- We adopt both perplexity and length ratios as crucial indicators for ICL example selection, contributing to the system's self-improvement mechanism.
- Our experiments show superior performance of the system on 5 low-resource languages across different LLM scales, demonstrating its versatility.

2 Related Work

LLMs have demonstrated remarkable capabilities across a range of natural language processing tasks, showcasing their potential to effectively tackle downstream machine translation tasks. Notably, these LLMs can achieve impressive performance with minimal or even no task-specific fine-tuning, a feature particularly advantageous for low-resource languages (Bawden and Yvon, 2023; Jiao et al., 2023). This capability is frequently attributed to advanced techniques such as prompt design and in-context learning.

Effective communication with AI systems requires practice and understanding of optimal interaction strategies. As such, automatic prompt optimization (Yang et al., 2024c; Wang et al., 2024) has emerged as an active area of research, with machine translation being no exception. Recent studies highlight significant variations in zero-shot prompting performance based on the template used (Zhang et al., 2023a). Additionally, it has been discovered that the stylistic elements of a prompt influences the quality of translation outputs (Jiao et al., 2023).

Turning to the field of in-context learning, the strategy for selecting demonstration examples plays a crucial role in performance outcomes. Research indicates that employing diverse strategies for prompt example selection can lead to varying results (Zhang et al., 2023a). Furthermore, some scholars argue that the intrinsic quality of an example often outweighs its proximity to the current source sentence in terms of importance (Vilar et al., 2023). Few-shot demonstrations have been shown to influence the output in terms of language variety and formality (Garcia et al., 2023). Efficient augmentation of multiple ICL prompt inputs has been found to enhance the accuracy and confidence of LLM predictions (Yao et al., 2023). Moreover, the accuracy of translations can vary significantly based on the examples included in the prompt (Merx et al., 2024): for instance, one-shot task-level example improves translation quality (Agrawal et al., 2023), and providing LLMs with specific examples or relevant contextual information about the translation task substantially improves their performance (Jiang and Zhang, 2024).

3 DAIL-translation Approach

Since we do not have enough data to fine-tune an LLM, DAIL-translation enhances translation capabilities through the integration of three interconnected components (Figure 1). For each language, the system maintains two databases—one for high-quality translations and another for potentially incorrect translations—alongside an instruction optimizer, a demonstration manager, and a quality estimator. The translation process for a single utterance involves four steps: (1) for a given query Q to be translated, we first check if there are enough number of wrong translations $|\{(b_q, b_t)\}|$ available. If so, the instruction optimizer refines the current translation instruction I to generate an improved instruction I'; otherwise, this step is bypassed. (2)

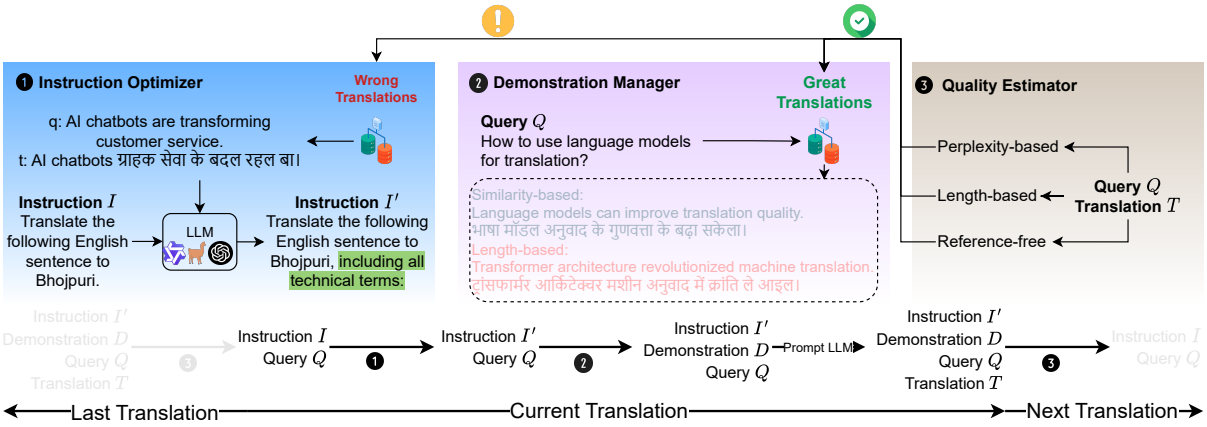

Figure 1: System architecture of our translation framework consisting of three main components: (1) Instruction Optimizer dynamically refines translation prompts based on failure cases, (2) Demonstration Manager that intelligently retrieves relevant examples through similarity and length-based matching, and (3) Quality Estimator that evaluates translation quality using perplexity-based, length-based, and reference-free metrics. The bottom timeline illustrates the system's life-long learning capability, where current translation contributes to continuous improvement - wrong translations set aids in prompt optimization, while successful ones facilitate ICL sampling in the future.

Demonstration Manager intelligently retrieves relevant examples D from $\{(g_q, g_t)\}$ through a combination of similarity-based search and length-ratio-based sampling to assist with the translation; (3) LLM takes Q, I' and D as inputs to produce translation output T; (4) Quality estimator evaluates the output quality of T given Q, and determines which database the translation should be stored into.

3.1 Instruction Optimizer

Large Language Models (LLMs) have been shown to be highly sensitive to the prompt format (Zhao et al., 2021). Notably, semantically similar prompts can yield drastically different performance outcomes (Kojima et al., 2022; Zhou et al., 2023; Zhang et al., 2023b). In some instances, optimized prompts may include several uninterpretable tokens (Wen et al., 2023), making it challenging for humans to discover and construct such effective prompts manually. Recent work (Yang et al., 2024c; Wang et al., 2024) has shown that LLMs can be utilized to optimize instruction, but this often involves repeatedly scoring the performance of historical prompts on the same dataset, which is time-consuming and costly. To address this issue, we propose dynamically refining translation prompts based on past failure cases. This approach is analogous to a self-reflective process (Shinn et al., 2023; Ji et al., 2023), where errors serve as the foundation for future enhancements.

The optimization process is conducted in a black-box manner, making it applicable to both open-source models and LLMs that are accessible only through API calls. In each optimization step, we provide the optimizer LLM with the instruction trajectory as contextual hints, current $\{(b_q, b_t)\}$ as semantic gradients, and a description of the optimization goal as well as how to utilize the provided information. It is important to note that the potentially wrong translations are removed following the completion of the optimization step. The prompt templates used for this process can be diverse, a sample of which is detailed in Appendix A.

3.2 Demonstration Manager

In-context parallel examples enhance machine translation by providing the model with knowledge of the task and the desired output format (Agrawal et al., 2023). It is well-established that selecting ICL examples based on cosine similarity outperforms random selection because it provides more contextually relevance to the previously unseen source sentence. However, in translation tasks involving low-resource languages, particularly when they are the source language, identifying multiple highly similar examples becomes challenging. This difficulty can stem from: (1) the limited availability of the parallel data from which to retrieve examples; and (2) the relatively weak tokenizer and embedding models for low-resource languages.

With the aim of identifying an efficient solution that complements traditional similarity-based methods, we draw inspiration from the Gale-Church alignment algorithm (Gale and Church, 1991; Liu

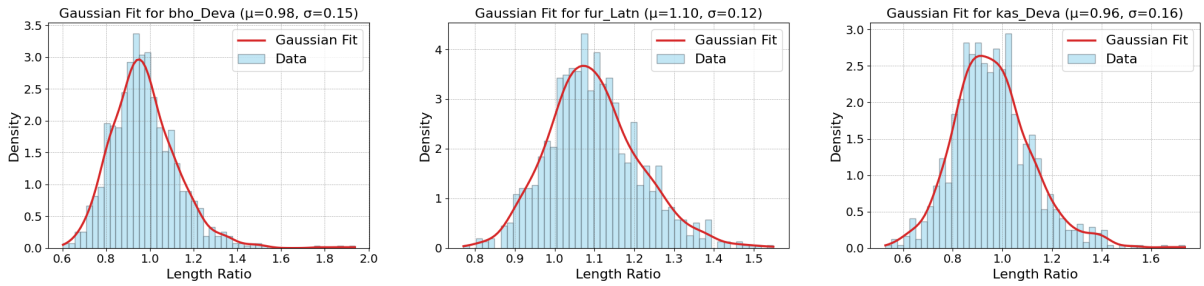

Figure 2: Fitted Gaussian distribution of length ratios for three different low-resource languages.

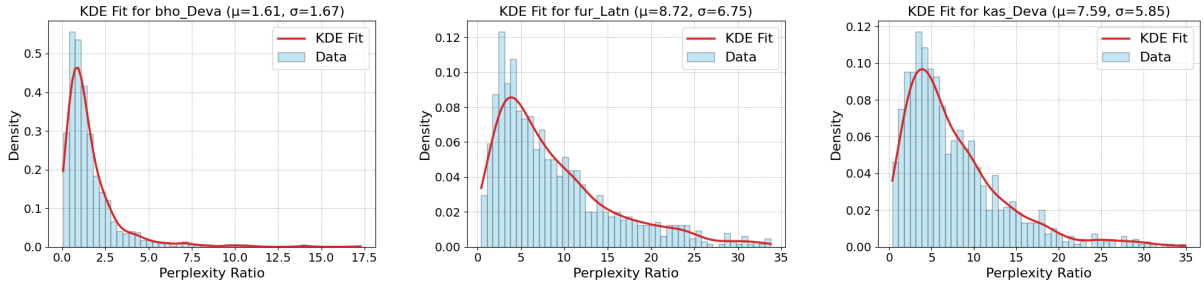

Figure 3: Fitted distribution of Perplexity Ratios for three different low-resource languages.

et al., 2024), which highlights that the character-level length ratio (length ratio for short in following texts) between source and target sentences typically varies around a fixed value, generally following a Gaussian distribution. Our analysis across ten datasets reveals that the length ratio between low-resource and English pairs conforms to a language-specific Gaussian distribution. This insight implies that if the parameter (*i.e.* mean and standard deviations) of this distribution can be determined, the desired target sentence length can be estimated from the source sentence. Accordingly, when selecting ICL examples, the Demonstration Manager utilizes this length ratio distribution to enhance the model's comprehension of the output format.

Parameter Estimation. The parameters of the distribution are determined by fitting them to $\{(g_q, g_t)\}$. This set can be cold-started using *pseudo parallel examples* (Zhang et al., 2023a), where we translate a collection of English sentences into low-resource languages via zero-shot prompting. Specifically, (1) We uniformly sample 1,000 English sentences from the dataset provided by (Maillard et al., 2023), who extracted sentences from Wikimedia's "List of articles every Wikipedia should have". *We reveal an increase in model performance when data gets larger*, and approximately 1,000 cold-start examples is enough to make a statistically significant improvement. (2) We translate them into low-resource languages with GPT-4o. This method simulates the common scenario where low-resource text is scarce, whereas English monolingual corpora are abundant and readily accessible. Since assumed distribution is Gaussian, length ratios that deviate beyond 3σ are considered outliers and removed for LLM-generated translation. The resulting distribution closely aligns with that of human translations. Consequently, we utilize data translated by LLMs as the initial $\{(g_q, g_t)\}$. The distribution for different languages are illustrated in Figure 2. For each query, we sample one example from $\{(g_q, g_t)\}$ according to the fitted distribution.

3.3 Quality Estimator

After each translation, the query-translation pair denoted as (Q, T) is allocated to one of $\{(b_q, b_t)\}$ or $\{(g_q, g_t)\}$. The assignment is determined by the Quality estimator, which evaluates the output quality to ascertain its appropriate position. This mechanism not only expands the ICL search space for the Demonstration Manager but also providing more bad cases for the Instruction Optimizer. In our experiment, the numbers of pairs in $\{(b_q, b_t)\}$ and $\{(g_q, g_t)\}$ are 30 and 970 respectively in the cold-start phase for `fur_Latn`; while the numbers in the self-improvement phase become 50 and 1962.

Like mentioned before, the length ratio could be served as crucial indicators for selection. If the length ratio is in-distribution, we arrange them to $\{(g_q, g_t)\}$, otherwise to $\{(b_q, b_t)\}$. Besides we

Table 1: BLEU and chrF++ scores for five low-resource languages (xxx-eng). **Bold** numbers denote the highest scores across all systems. Statistical significance compared to the second-best system is indicated by dark blue ($p < 0.05$) and dark yellow ($p >= 0.05$), computed using paired bootstrap resampling (Koehn, 2004). Note that for token efficiency, we only compare Zero-shot GPT-4o with our model.

LLM	Method	fur_Latn		lij_Latn		lmo_Latn		bho_Deva		hne_Deva	
		BLEU↑	chrF++↑	BLEU↑	chrF++↑	BLEU↑	chrF++↑	BLEU↑	chrF++↑	BLEU↑	chrF++↑
Qwen-2_{7B}	Zero-shot	17.7	43.0	21.4	46.5	20.0	44.5	12.2	37.7	17.0	43.7
	CoT	16.5	40.8	18.5	42.7	19.1	43.4	11.3	35.7	15.2	41.1
	ICL	22.8	47.5	25.7	50.5	25.0	49.1	12.6	38.3	17.0	43.6
	CoT & ICL	23.1	47.8	25.9	50.7	25.4	49.5	14.1	39.9	19.7	45.9
	Ours	**23.3**	**48.0**	**26.3**	**51.0**	**25.5**	**49.6**	**14.7**	**40.7**	**20.1**	**46.5**
	p-value	0.007	0.012	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Qwen-2_{72B}	Zero-shot	26.7	50.5	28.5	52.1	26.9	50.2	16.3	40.5	24.3	48.8
	CoT	24.2	47.2	24.9	46.7	23.7	45.6	13.1	33.9	22.4	45.5
	ICL	33.3	56.9	37.0	59.9	34.3	57.6	22.0	47.6	29.9	54.6
	CoT & ICL	32.8	56.4	36.5	59.4	34.2	57.6	22.3	47.7	30.1	54.6
	Ours	**34.5**	**58.4**	**38.0**	**61.0**	**35.3**	**58.7**	**23.5**	**49.4**	**31.0**	**56.1**
	p-value	0.000	0.070	0.000	0.000	0.000	0.005	0.000	0.000	0.000	0.000
LLAMA-3_{8B}	Zero-shot	24.7	49.6	26.4	51.1	23.2	47.6	15.5	42.0	18.0	45.1
	CoT	23.9	48.6	25.3	50.0	21.4	45.9	13.9	39.7	17.0	43.7
	ICL	31.7	55.5	32.8	56.4	29.8	54.1	19.1	45.3	23.2	49.5
	CoT & ICL	31.6	55.5	32.8	56.4	29.5	53.7	19.1	45.3	23.2	49.5
	Ours	**32.2**	**56.0**	**33.6**	**57.2**	**30.5**	**54.8**	**19.9**	**46.2**	**23.8**	**49.9**
	p-value	0.000	0.000	0.000	0.000	0.000	0.000	0.083	0.177	0.090	0.041
LLAMA-3_{70B}	Zero-shot	32.8	57.3	33.7	57.7	27.5	51.4	19.6	46.5	25.0	51.6
	CoT	31.8	56.4	32.8	56.9	19.4	40.3	18.9	45.6	24.3	50.9
	ICL	39.9	62.8	41.1	63.3	37.1	60.0	25.0	51.2	32.7	57.9
	CoT & ICL	39.9	62.8	41.0	63.3	35.9	58.9	25.1	51.4	33.0	58.4
	Ours	**40.1**	**63.1**	**41.4**	**63.6**	**38.0**	**60.8**	**25.7**	**51.9**	**33.9**	**59.1**
	p-value	0.170	0.184	0.043	0.002	0.026	0.378	0.003	0.292	0.001	0.203
GPT-4o	Zero-shot	41.2	64.2	43.1	65.4	40.4	63.1	29.5	55.2	42.4	66.2
	Ours	**44.1**	**66.1**	**47.0**	**68.1**	**43.7**	**65.6**	**32.1**	**57.1**	**46.0**	**68.8**

also discover that the perplexity ratio between Q and T should be constrained within a certain range. Perplexity of a sentence can be defined as:

$$\text{Perplexity}(x) = \exp\left(\frac{1}{N}\sum_{i=1}^{N} -\log P(x_i|x_{<i})\right), \quad (1)$$

where x is a sequence of tokens of length N. In contrast to the approach proposed by (Liu et al., 2024), who model the perplexity ratio as following a Gaussian distribution, our analysis reveals a different understanding of this model-specific metric. Experimenting on LLAMA-3_{8B} (Dubey et al., 2024), we observed that the distribution of perplexity ratios exhibits a severely left-skewed Gaussian shape, as illustrated in Figure 3. This discovery has one important implication: The observed distribution provides a potential upper bound on expected perplexity ratio values. This upper limit serves as a valuable constraint in selection mechanisms, allowing for more informed decision-making processes when evaluating model outputs. Hence it is important to emphasize that while our findings challenge the assumption of a standard Gaussian distribution, they do not diminish the utility of perplexity ratios as indicators for selection tasks. On the contrary, the understanding enhances their potential as discriminative features by providing a more accurate representation of their behavior. The combination of these two metrics allows for a more robust selection mechanism that can account for both content and form variations. Several distinct categories of translation errors that could be effectively identified by the quality estimator are shown in Appendix B.

4 Experiments

In this section, we present analysis of results using automatic evaluation metrics against 5 low-resource languages from the FLORES-200 benchmark (Costa-jussà et al., 2022).

4.1 Data and Large Language Models

Following (Maillard et al., 2023), we select two distinct clusters of related languages to investigate the efficacy of our proposed approach across different linguistic families and script systems. The first

Table 2: BLEU and chrF++ scores for five low-resource languages (eng-xxx). **Bold** numbers denote the highest scores across all systems. Statistical significance compared to the second-best system is indicated by dark blue ($p < 0.05$) and dark yellow ($p >= 0.05$), computed using paired bootstrap resampling (Koehn, 2004). Note that for token efficiency, we only compare Zero-shot GPT-4o with our model.

LLM	Method	fur_Latn		lij_Latn		lmo_Latn		bho_Deva		hne_Deva	
		BLEU↑	chrF++↑	BLEU↑	chrF++↑	BLEU↑	chrF++↑	BLEU↑	chrF++↑	BLEU↑	chrF++↑
Qwen-2_{7B}	Zero-shot	2.7	24.3	1.9	23.0	1.7	18.4	2.3	20.1	3.9	23.9
	CoT	3.0	25.4	2.3	24.0	1.9	19.0	2.7	20.7	4.0	24.5
	ICL	3.9	24.5	3.4	27.6	3.3	21.7	3.2	20.3	4.8	26.4
	CoT & ICL	3.3	22.9	4.1	28.9	3.1	21.2	3.3	20.4	5.0	26.6
	Ours	**5.0**	**29.0**	**4.1**	**29.2**	**3.8**	**25.1**	**3.6**	**23.1**	**5.1**	**26.8**
	p-value	0.000	0.000	0.000	0.000	0.019	0.000	0.113	0.000	0.000	0.028
Qwen-2_{72B}	Zero-shot	3.8	27.9	3.8	28.4	4.1	27.1	4.7	25.3	5.6	27.3
	CoT	3.8	27.6	3.7	28.0	4.2	26.4	4.0	21.3	5.3	24.2
	ICL	7.0	30.3	5.1	30.2	4.2	24.9	6.0	27.6	7.6	30.1
	CoT & ICL	6.2	27.8	5.3	30.5	3.7	23.1	6.1	27.5	7.6	30.2
	Ours	**8.1**	**34.1**	**5.4**	**30.9**	**5.1**	**28.9**	**6.8**	**29.4**	**8.8**	**33.8**
	p-value	0.000	0.455	0.008	0.000	0.014	0.001	0.060	0.102	0.000	0.000
LLAMA-3_{8B}	Zero-shot	6.1	32.3	4.0	28.1	3.2	24.7	4.5	26.0	5.3	28.9
	CoT	6.0	32.2	4.0	27.2	3.2	23.8	4.5	26.0	5.6	29.2
	ICL	9.3	32.8	6.5	31.5	4.0	22.6	4.8	23.4	6.9	28.5
	CoT & ICL	9.6	33.1	6.9	32.0	3.4	20.4	5.1	24.1	7.9	32.4
	Ours	**12.0**	**38.3**	**7.2**	**32.8**	**5.9**	**29.2**	**6.9**	**30.2**	**8.4**	**33.3**
	p-value	0.000	0.000	0.196	0.000	0.000	0.000	0.000	0.000	0.116	0.003
LLAMA-3_{70B}	Zero-shot	21.1	48.7	9.6	37.9	6.4	32.8	8.8	34.4	7.1	32.7
	CoT	20.8	48.2	9.7	37.9	6.4	32.7	8.7	34.1	7.1	32.6
	ICL	22.9	49.5	11.4	40.1	7.0	32.2	11.5	36.7	12.5	40.1
	CoT & ICL	23.4	49.4	11.4	40.0	6.9	32.1	11.6	36.6	12.7	40.2
	Ours	**23.9**	**50.3**	**11.6**	**40.5**	**7.3**	**33.6**	**11.7**	**37.7**	**13.6**	**41.3**
	p-value	0.000	0.000	0.046	0.000	0.000	0.005	0.001	0.001	0.000	0.000
GPT-4o	Zero-shot	20.6	46.8	8.8	36.7	7.4	**34.2**	13.7	**41.0**	13.7	41.3
	Ours	**23.0**	**49.6**	**9.8**	**38.7**	**7.6**	33.6	**13.9**	**41.0**	**15.6**	**44.4**

cluster comprises three languages from the Italic branch (`fur_Latn`, `lij_Latn`, `lmo_Latn`), written in Latin script; The second cluster focuses on four languages from the Indo-Aryan branch written in Devanagari script (`bho_Deva`, `hne_Deva`). Each language dataset has 1012 samples. Our experimental design adopts an English-centric approach: specifically, we focus on two primary translation directions: (1) xxx-eng: Translation from any of the selected languages to English; (2) eng-xxx: Translation from English to any of the selected languages. More details can be found in Appendix C.

LLMs are specifically selected to represent different scales, architectures, and training paradigms to ensure a broad and representative assessment. The models chosen are: (1) LLAMA-3_{8B} and LLAMA-3_{70B} (Dubey et al., 2024); (2) Qwen-2_{7B} and Qwen-2_{72B} (Yang et al., 2024a); (3) GPT-4o (Hurst et al., 2024) accessed through API.

4.2 Baselines

We consider the following comparisons:

- **Zero-shot**: LLMs are prompted without any additional aids or context.
- **Chain-of-thought** (Wei et al., 2022) (CoT): LLMs are provided with instructions that encourage step-by-step reasoning, specifically, "please think step by step" is added to the end of the prompt.
- **ICL** (Brown et al., 2020): LLMs are prompted with 5 exemplars of successful translations.
- **CoT & ICL**: Combination of CoT and ICL.

4.3 Results

To evaluate the effectiveness of DAIL-translation, we conduct experiments on five language pairs in both xxx-eng and eng-xxx directions across different LLM architectures. The results are shown in Tables 1 and 2. Note that the number of ICL examples for all few-shot methods including "ours" are equal to 5. The sentence representations are obtained with *Qwen-text-embedding-v3 model*. The statistical significance is computed using paired bootstrap resampling (Koehn, 2004): we test on 1,000 different test sets to reduce estimation error, where the test sets are generated by randomly sam-

pling queries with replacement from the original test collection. We have following observations: (1) For both xxx-eng and eng-xxx translations, DAIL-translation consistently outperforms all baseline approaches. Our method consistently achieves the highest BLEU and chrF++ scores, and is statistically significant compared to the second-best systems in most of the cases. This suggests that our approach is particularly effective at handling the challenges posed by low-resource language translation, providing superior translation quality compared to existing methods. (2) DAIL-translation improves the translation performance in the Qwen and LLAMA family across different parameter sizes indicates that our method effectively enhances the model's cross-lingual transfer capabilities regardless of the underlying architecture. (3) Comparing CoT with Zero-shot, we reveal an interesting finding: CoT yields only marginal improvements or even decreases performance. While CoT has proven beneficial in many NLP tasks, these results suggest it may not be well-suited for low-resource translation tasks.

4.4 Computational Overhead

As we introduce three modules to our system, we estimate the computational overhead as follows:

- **Perplexity Estimation** needs a one-pass inference using LLAMA-3_{8B}, which incurs little cost.
- **Dynamic Demonstration Selection** involves cosine similarity between the query and stored demonstrations, which is computationally lightweight.
- **Prompt Optimization** is executed only when detecting enough problematic cases based on out-of-distribution length-ratio and perplexity ratio samples. These problematic cases are systematically removed post-optimization, reducing frequency and computational needs over time.

5 Analysis

In this section, we present analysis of results from the perspective of the instruction optimization process; and investigate if our system enhances the model's ability to understand the length format of the translation pairs. More experimental results are shown in Appendix D.

5.1 Are optimized instructions transferable?

A crucial part of the system, mentioned in § 3.1, is the instruction optimizer. In this section, we would like to dive deep into the properties of the optimized instructions. Based on Table 3, our findings reveal several key observations:

Table 3: Performance changing when xxx-eng direction prompt is optimized on `fur_Latn` for gpt-4o. Shallow blue / Shallow yellow indicates the translation performance increases / decreases after optimization.

Data	chrF++		BLEU	
	before	after	before	after
`fur_Latn`	46.8	48.8	20.6	21.6
`lij_Latn`	38.7	38.0	8.8	8.6
`lmo_Latn`	34.2	33.8	7.4	7.4
`bho_Deva`	41.0	39.0	13.9	12.7
`hne_Deva`	41.3	37.0	13.7	11.1

First, the instruction optimization process improves the performance for the source language (`fur_Latn`), with absolute gains of +2.0 and +1.0 points in chrF++ and BLEU scores, respectively. This confirms the effectiveness of our optimization approach within the target domain.

However, this improvement may not be transferable. We can observe consistent performance degradation when applying the optimized prompt to other languages, with varying degrees. Notably, languages sharing the Latin script (`lij_Latn`, `lmo_Latn`) show relatively minor degradation (-0.1 and -0.55 on average for BLEU and chrF++ scores respectively); In contrast, languages utilizing the Devanagari script (`bho_Deva`, `hne_Deva`) demonstrate significant performance drops (-1.4 and -3.15 respectively), indicating potential script-specific barriers about prompt optimization. These findings have important implications for multilingual prompt optimization strategies.

5.2 How to choose the language of prompt?

Understanding the optimal prompt language is crucial for effective machine translation, especially when dealing with low-resource languages using different scripts. To investigate this, we analyze the language of prompts in our translation setup across different language pairs and directions. Table 4 shows the number of English prompts out of 20 total prompts for different translation directions, the 20 prompts consists of the best-5 prompts for 4 open-source LLMs. For xxx-eng direction, all prompts are consistently in English regardless of the source language. However, for eng-xxx di-

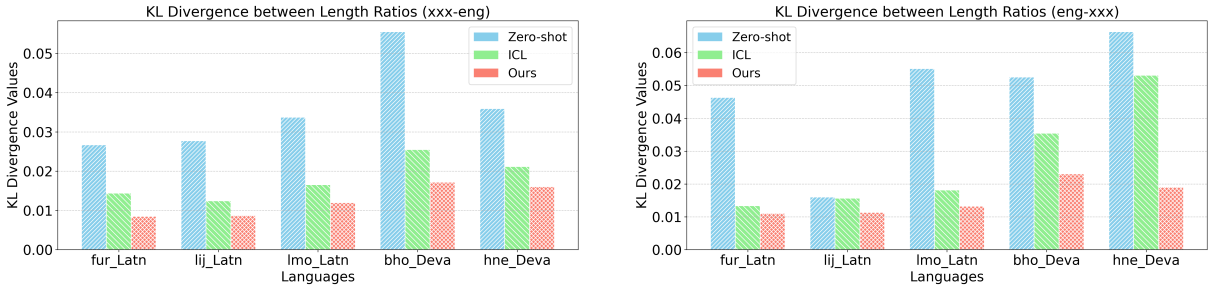

Figure 4: KL Divergence between Length Ratio Distributions of Ground Truth and Three Methods.

rection, the prompt language depends on the target script: languages using Latin script maintain English prompts, while languages using Devanagari script require prompts in their respective target languages, resulting in very few English prompts. This finding is further supported by empirical results in Figure 5 where translating to hne_Deva achieved better performance (13.6 BLEU) when using prompts in Devanagari script compared to English prompts (12.5 BLEU), suggesting that matching the prompt script to the target language is beneficial for Devanagari script languages. Besides language type, we also provide with a prompt optimization case study in Appendix E.

5.3 Are length ratios getting better?

To evaluate whether our system improves the handling of length relationships, we employ KL divergence to measure how closely the length ratio distributions match between different translation methods. The experimental results demonstrate several notable patterns in KL divergence across different language pairs and translation methods: (1) The proposed method achieves the lowest KL divergence values consistently, followed by ICL, while Zero-shot shows the highest divergence. This indicates that ICL can enhance the model's ability to understand formats, and select ICL examples based on length ratio further improves this desired property. (2) Devanagari script languages exhibit higher divergence compared to Latin script languages, possibly due to greater structural differences from English which is also written in Latin. (3) We can observe directional asymmetry, where eng-xxx translations show slightly higher divergence than xxx-eng, which is consistent with the previous findings, who have shown that current LLMs are most effective at machine translation when English is the target language (Enis and Hopkins, 2024; Zhu et al., 2024b) (*i.e.* they are better at xxx-eng translation than eng-xxx translation).

Table 4: Distribution of English prompts across different translation directions and languages. The total 20 prompts consists of the best-5 prompts for 4 open-source LLMs. For instance, when translating from English to hne_Deva, none of the prompts are in English.

Data	fur_Latn	lij_Latn	lmo_Latn	bho_Deva	hne_Deva
xxx-eng	20/20	20/20	20/20	20/20	20/20
eng-xxx	17/20	17/20	20/20	1/20	0/20

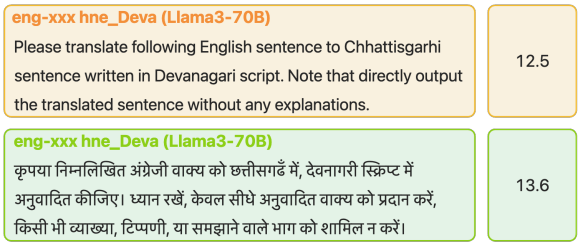

Figure 5: Translation prompts in source and target language with their respective BLEU scores for English-to-Chhattisgarhi translation using LLAMA-3_{70B}.

6 Conclusion

In this paper, we propose DAIL-translation, a system to improve the translation ability of LLMs in endangered languages with minimal cost. Our system consists of three components: Instruction Optimizer, Demonstration Manager, and Quality Estimator. Starting from 1k monolingual English sentences, our system achieves good performance through self-improving on 5 low-resource languages across different LLM parameter scales.

During investigation, we discover that optimized instructions are language-specific with limited transferability; however, there may exist a script-dependent transfer pattern which helps generalization. For Devanagari script languages, matching the prompt script to the target language can be beneficial. Our finding shows that length-based parallel example selection can provide a complementary advantage to similarity-based searching by enhancing the model's ability to understand formats.

7 Limitations

The major limitation of our experiments is evaluation type. Because the languages that we work with in this paper are low-resource, it was not feasible to find native speakers to do human evaluation (at least for us) on the output of our models. Furthermore, previous work (Xu et al., 2024) has shown that metrics like BLEU may focus on lexical matches but lack semantic depth; however, reference-free evaluation models such as XCOMET (Guerreiro et al., 2024) and KIWI-XXL (Rei et al., 2022) doesn't support low-resource languages used in the paper, hence we don't do reference-free evaluation.

References

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

Rachel Bawden and François Yvon. 2023. Investigating the translation performance of a large multilingual language model: the case of BLOOM. In *EAMT*, pages 157–170. European Association for Machine Translation.

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. In *NeurIPS*.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Y. Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *CoRR*, abs/2207.04672.

Sara Court and Micha Elsner. 2024. Shortcomings of llms for low-resource translation: Retrieval and understanding are both the problem. In *WMT*, pages 1332–1354. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Maxim Enis and Mark Hopkins. 2024. From LLM to NMT: advancing low-resource machine translation with claude. *CoRR*, abs/2404.13813.

William A. Gale and Kenneth Ward Church. 1991. A program for aligning sentences in bilingual corpora. In *ACL*, pages 177–184. ACL.

Xavier Garcia, Yamini Bansal, Colin Cherry, George F. Foster, Maxim Krikun, Melvin Johnson, and Orhan Firat. 2023. The unreasonable effectiveness of few-shot learning for machine translation. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 10867–10878. PMLR.

Nuno Miguel Guerreiro, Ricardo Rei, Daan van Stigt, Luísa Coheur, Pierre Colombo, and André F. T. Martins. 2024. xcomet : Transparent machine translation evaluation through fine-grained error detection. *Trans. Assoc. Comput. Linguistics*, 12:979–995.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex

Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll L. Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, and Dane Sherburn. 2024. Gpt-4o system card. *CoRR*, abs/2410.21276.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating LLM hallucination via self reflection. In *EMNLP (Findings)*, pages 1827–1843. Association for Computational Linguistics.

Zhaokun Jiang and Ziyin Zhang. 2024. Can chatgpt rival neural machine translation? A comparative study. *CoRR*, abs/2401.05176.

Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. 2023. Is chatgpt A good translator? A preliminary study. *CoRR*, abs/2301.08745.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. 2017. Bag of tricks for efficient text classification. In *EACL (2)*, pages 427–431. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *EMNLP*, pages 388–395. ACL.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *NeurIPS*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *SOSP*, pages 611–626. ACM.

Hengjie Liu, Ruibo Hou, and Yves Lepage. 2024. High-quality data augmentation for low-resource NMT: combining a translation memory, a GAN generator, and filtering. *CoRR*, abs/2408.12079.

Jean Maillard, Cynthia Gao, Elahe Kalbassi, Kaushik Ram Sadagopan, Vedanuj Goswami, Philipp Koehn, Angela Fan, and Francisco Guzmán. 2023. Small data, big impact: Leveraging minimal data for effective machine translation. In *ACL (1)*, pages 2740–2756. Association for Computational Linguistics.

Raphaël Merx, Aso Mahmudi, Katrina Langford, Leo Alberto de Araujo, and Ekaterina Vylomova. 2024. Low-resource machine translation through retrieval-augmented LLM prompting: A study on the mambai language. *CoRR*, abs/2404.04809.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*.

Ricardo Rei, José G. C. de Souza, Duarte M. Alves, Chrysoula Zerva, Ana C. Farinha, Taisiya Glushkova, Alon Lavie, Luísa Coheur, and André F. T. Martins. 2022. COMET-22: unbabel-ist 2022 submission for the metrics shared task. In *WMT*, pages 578–585. Association for Computational Linguistics.

Nathaniel R. Robinson, Perez Ogayo, David R. Mortensen, and Graham Neubig. 2023. Chatgpt MT: competitive for high- (but not low-) resource languages. In *WMT*, pages 392–418. Association for Computational Linguistics.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *NeurIPS*.

Inacio Vieira, Will Allred, Séamus Lankford, Sheila Castilho, and Andy Way. 2024. How much data is enough data? fine-tuning large language models for in-house translation: Performance evaluation across multiple dataset sizes. In *AMTA (1)*, pages 236–249. Association for Machine Translation in the Americas.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George F. Foster. 2023. Prompting palm for translation: Assessing strategies and performance. In *ACL (1)*, pages 15406–15427. Association for Computational Linguistics.

Xinyuan Wang, Chenxi Li, Zhen Wang, Fan Bai, Haotian Luo, Jiayou Zhang, Nebojsa Jojic, Eric P. Xing, and Zhiting Hu. 2024. Promptagent: Strategic planning with language models enables expert-level prompt optimization. In *ICLR*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2023. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. In *NeurIPS*.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. In *ICML*. OpenReview.net.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024a. Qwen2 technical report. *CoRR*, abs/2407.10671.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024b. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *CoRR*, abs/2409.12122.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2024c. Large language models as optimizers. In *ICLR*. OpenReview.net.

Bingsheng Yao, Guiming Chen, Ruishi Zou, Yuxuan Lu, Jiachen Li, Shao Zhang, Sijia Liu, James A. Hendler, and Dakuo Wang. 2023. More samples or more prompt inputs? exploring effective in-context sampling for LLM few-shot prompt engineering. *CoRR*, abs/2311.09782.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023a. Prompting large language model for machine translation: A case study. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 41092–41110. PMLR.

Tianjun Zhang, Xuezhi Wang, Denny Zhou, Dale Schuurmans, and Joseph E. Gonzalez. 2023b. TEMPERA: test-time prompt editing via reinforcement learning. In *ICLR*. OpenReview.net.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *ICLR*. OpenReview.net.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024a. Multilingual machine translation with large language models: Empirical results and analysis. In *NAACL-HLT (Findings)*, pages 2765–2781. Association for Computational Linguistics.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024b. Multilingual machine translation with large language models: Empirical results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2765–2781.

A Prompt Optimization Template

Objective: Optimize system prompts for high-quality translation in a low-resource language setting.
Optimization Framework: 1. Prompt Variation Methodology
- Generate diverse prompt variations
- Systematically modify:
a) Role specification
b) Instruction clarity
c) Contextual examples
d) Linguistic guidance
2. Evaluation Criteria
- BLEU score (0-100)
- Chrf++ score (0-100)
3. Iteration Strategy
- Analyze current top-performing prompts
- Identify common successful patterns
- Generate new prompts building on these insights
4. Challenging Case Analysis
- Catalog translation difficult cases
- Use bad cases to inform prompt refinement
- Create targeted variations addressing specific challenges
Deliverables:
- Ranked prompt variations
- Detailed performance breakdown
- Insights into prompt design effectiveness
Previous system prompts are arranged in ascending order based on their bleu scores, where higher scores indicate better quality.
$\{prompt\_with\_scores\}$
Here is a list of challenging cases for the given prompts:
$\{challenge\_cases\}$
Write your new text that is different from the old ones and has a score as high as possible.

B Failed Examples

To confirm that improvements are indeed due to the proposed heuristics rather than general repeated optimization, failed examples are selected randomly. Our quality metrics effectively identify several distinct categories of translation errors in Table 5. We believe the proposed metrics complement each other in identifying different types of translation failures, providing a robust automated quality assessment system.

C Reproducibility

C.1 Evaluation Metrics

We evaluate translation quality using ChrF++ and BLEU metrics via sacrebleu[1]. The signatures are: `chrF2++|nrefs:1|case:mixed|eff:yes|nc:6|nw:2|space:no|version:2.5.1` and `bleu|nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.5.1`.

C.2 Experimental Setup

To ensure efficient and scalable inference, we deploy our selected LLMs on up to 4 GPUs using vLLM serving system (Kwon et al., 2023). The sampling parameters involves a temperature of 0.0, a maximum output length of 200 tokens, with stop tokens as "$< |eot\_id| >$" and "$< |start\_header\_id| >$".

D Ablation Study

In our paper, the quality estimator and the instruction optimizer are interdependent. The quality estimator provides the potentially bad case to instruction optimizer, and the instruction optimizer relies on these bad cases to do prompt optimization. Hence we conduct the ablation study without Demonstration Manager in Table 6.

E Prompt Optimization Case Study

To answer the question about *What kind of prompts are much better than others?* We provide with an example of prompt optimization steps.

Initial: You are an expert linguist specializing in rare and endangered languages. Please translate following English sentence to Friulian sentence written in Latin script.

Step 1 (Format): You are an expert linguist specializing in rare and endangered languages. Please translate following English sentence to Friulian sentence written in Latin script. Note that directly output the translated sentence without any explanations.

Step 2 (Language specificity): You are an expert translator specializing in Friulian. Translate the following English sentence into Friulian, avoiding influences from Italian languages. Use genuine Friulian vocabulary, expressions, and grammatical structures. Maintain the original sentence structure where possible, but prioritize natural Friulian

[1] https://github.com/mjpost/sacrebleu

Table 5: Detection Methods and Their Associated Issues

Detection Method	Issue Category	Specific Cases and Results
Length Ratio	Token Generation	Missing token Repetitive/circular output patterns Result: Abnormally long translations
	Content Accuracy	Hallucinations (longer translations) Content omissions (shorter translations)
Perplexity Ratio	Model Behavior	Response refusal Missing token with repetitive patterns Result: Unusually low perplexity scores
	Language Coherence	Mixed language output Result: Abnormally high perplexity scores
	Content Accuracy	Hallucinations Result: Elevated perplexity scores

Table 6: Translation performance (BLEU/Chrf++) across different language pairs without Demonstration Manager

Model	Direction	fur_Latn	lij_Latn	lmo_Latn	bho_Deva	hne_Deva
Qwen-2_{7B}	xxx-eng	17.7/43.0	21.4/46.5	20.0/44.5	12.2/37.7	17.0/43.7
Qwen-2_{7B}-Enhanced	xxx-eng	18.6/43.9	21.5/46.8	20.8/45.4	13.3/39.3	17.8/44.7
LLAMA-3_{8B}	xxx-eng	24.7/49.6	26.4/51.1	23.2/47.6	15.5/42.0	18.0/45.1
LLAMA-3_{8B}-Enhanced	xxx-eng	26.6/51.5	26.5/51.3	24.8/49.6	16.6/43.3	20.0/47.5
Qwen-2_{7B}	eng-xxx	2.7/24.3	1.9/23.0	1.7/18.4	2.3/20.1	3.9/23.9
Qwen-2_{7B}-Enhanced	eng-xxx	3.5/26.2	3.8/28.4	2.6/19.2	3.5/22.6	4.3/25.5
LLAMA-3_{8B}	eng-xxx	6.1/32.3	4.0/28.1	3.2/24.7	4.5/26.0	5.3/28.9
LLAMA-3_{8B}-Enhanced	eng-xxx	7.3/33.9	4.6/28.4	4.1/26.7	5.2/27.4	6.3/30.8

phrasing. Note that directly output the translated sentence without any explanations.

Step 3 (Comprehensive guidelines): You are an expert translator specializing in Friulian. Translate the following English sentence into Friulian, avoiding influences from Italian languages. Use genuine Friulian vocabulary, expressions, and grammatical structures. Maintain the original sentence structure where possible, but prioritize natural Friulian phrasing. For technical or specialized terms, use established Friulian equivalents if they exist, or create appropriate neologisms based on Friulian linguistic patterns. Preserve all named entities, numbers, and punctuation. Aim for a translation that a native Friulian speaker would produce, even for complex or technical content. Note that directly output the translated sentence without any explanations.