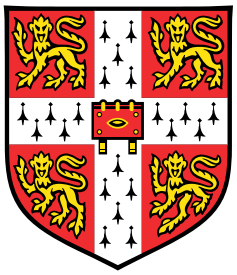
Multilingual Models in Neural Machine Translation



Guangyu Yang

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy in Machine Learning and Machine Intelligence

Clare Hall

August 2023

Dedicated to my loving parents.

Declaration

I, Guangyu Yang of Clare Hall, being a candidate for the MPhil in Machine Learning and Machine Intelligence, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

All software used in this thesis was written in Python and PyTorch. The open-source Hugging Face transformers library[1] [47] was used for downloading pre-trained models and inference. The sacreBLEU library[2] [26] was used to download datasets and evaluate the performance of the models using the BLEU metric. The BLEURT library[3] [34] was used to evaluate the performance of the models using the BLEURT metric. The OpenICL framework was used to experiment with in-context learning. And the Direct Preference Optimization codebase[4] [28] was used for unsupervised preference optimization with MBR. The remaining software was written by me from scratch with standard Python packages. Codes are availabel here.

The word count, including tables, footnotes, figure captions, and appendices, is 14582.

Guangyu Yang
August 2023

[1] https://github.com/huggingface/transformers
[2] https://github.com/mjpost/sacrebleu
[3] https://github.com/google-research/bleurt
[4] https://github.com/eric-mitchell/direct-preference-optimization

Acknowledgements

Firstly, I would like to thank my supervisor, Professor Bill Byrne, for proposing this project, and guiding me to explore this interesting topic throughout my thesis journey. His patience and invaluable insights helped me keep moving forward and illuminated the path of professional research for me. I am truly grateful to have him as my supervisor.

Secondly, I would like to thank my co-supervisor, Eric Chen, for helping me design the experiments and refine the dissertation. Every discussion with him deepened my understanding and perspective on the project.

Finally, I would like to thank my parents and all my friends at Cambridge, who unceasingly supported and encouraged me throughout this year.

Abstract

Multilingual Large Language Models (MLLMs) have undergone rapid development in recent years, following the success of monolingual Large Language Models (LLMs). Unlike previous Neural Machine Translation (NMT) models that rely on abundant parallel texts for training, MLLMs are able to perform machine translation using only a few translation pairs via in-context learning. And training the multilingual models only requires unparalleled monolingual datasets which are more accessible. Therefore, MLLMs show great potential for the task of NMT.

In this project, we investigate the translation performance of MLLMs by exploring several design choices of the in-context learning framework, focusing on the performance of different decoding methods. We show that Minimum Bayes Risk (MBR) decoding results in superior performance than widely used MAP decoding methods like beam search. However, MBR decoding is a two-pass algorithm which is slow and complex, thus causing high computational cost during inference.

To address the inherent complexity of MBR decoding, we propose a novel unsupervised preference optimization method that teaches the model to learn the preference of MBR decoding. The method consists of a data generation stage and a training stage. We explore different methods for constructing the preference dataset during the first stage, and investigate the effects of two different training objectives. We show that, under proper settings, MLLMs that are optimized for MBR preference achieve significantly better translation performance when decoded with beam search. Our results show the potential of unsupervised preference optimization that allows MLLMs to use fast MAP decoding methods while achieving the superior performance of MBR decoding.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Neural machine translation (NMT) has become the dominant paradigm in machine translation, leveraging deep neural networks to translate texts between languages [16, 7, 3, 40]. Previous NMT methods train specialized translation systems in a supervised fashion. These systems are able to generate faithful translations given a large corpus of translation pairs to learn from. However, the dependency on supervised training data and the model size limits the flexibility of such supervised systems. Recently, with the success of Large Language Models (LLMs) [6, 8, 42, 43], another line of work utilizes large language models and Multilingual Large Language Models (MLLMs) for machine translation and has shown that these large models possess impressive machine translation capabilities [12, 19, 53, 54]. The advantage of this new method is that it allows a single model to translate between multiple languages using a shared lexicon, with few or possibly even zero amount of supervised data.

Unlike previous supervised translation systems trained specifically on translation pairs, LLMs and MLLMs are only pre-trained on monolingual texts. During inference time, these models are able to generate translation of a source sentence conditioned on the source sentence itself and a few translation pairs as demonstration examples. This paradigm of in-context few-shot learning was first proposed by Brown et al. [6] and has been widely applied to Natural Language Generation (NLG) tasks. Recent works have shown the effectiveness of these models in machine translation task [12, 19, 53, 45, 54, 1].

Despite the promising results gained from using in-context learning for machine translation, few works have systematically explored the effect of decoding methods used for generating translations. Since LLMs and MLLMs have never been trained for translation in an end-to-end fashion, simply applying the commonly used decoding methods like greedy decoding and beam search might not be optimal. This motivates us to investigate an important research question of finding better suited decoding methods for the new in-context NMT paradigm.

In the first part of the project, we investigate the effectiveness of in-context few-shot learning for machine translation using MLLMs. We use existing MLLMs like XGLM [19] and BLOOM[1] to explore the effects of several key component of in-context learning with a focus on the decoding methods. We confirm that Minimum Bayes Risk (MBR) decoding [18] is a promising decoding method that outperforms other commonly used methods. MBR selects the best output from a list of translation hypotheses based on the Bayesian risk of each hypothesis. We systematically investigated the behavior of MBR decoding and found that MBR is able to provide ranking of translation hypotheses that has strong correlation with ranking based on the reference translation using an appropriate measure like BLEURT [34].

Although MBR decoding has advantages over commonly used decoding methods, the main drawback of MBR decoding is that it is a two-pass algorithm, which is slow and complex. This brings up our second research question of how to utilize the power of MBR without sacrificing inference time. In the second part of the project, we approach this question by proposing a novel unsupervised fine-tuning procedure that teaches the model to learn from MBR behavior. The fine-tuning is a form of preference optimization method based on Direct Preference Optimization (DPO) [28], where the MLLM learns the preference of MBR decoding, given a pair of translation candidates. After the fine-tuning stage, the model is ideally able to generate translations preferred by MBR with a single-pass decoding method like greedy decoding and beam search. Based on our findings in the first part of the project, we find that this unsupervised preference optimization method is a promising fine-tuning strategy.

To summarize, our contributions are as follows:

- We carry out systematic investigation on the key components for in-context NMT with MLLMs, and replicate the findings from recent works on the importance of demonstration examples. We show that the number of demonstration examples improves the performance up to a certain number, and focusing on the similarity between examples and query only results in small performance gain.
- We evaluate the performance of different decoding methods for in-context NMT and show that MBR decoding significantly improves translation performance of MLLMs.
- We propose a novel fine-tuning strategy that teaches the model to learn the preference of MBR decoding. To the best of our knowledge, we are the first to use unsupervised decoding preferences as the reward signal for preference optimization for NMT with MLLMs. The approach aims to overcome the computational complexities of MBR decoding, enabling efficient high-quality translation with standard decoding methods.

[1] https://bigscience.huggingface.co/blog/bloom

Chapter 2

Background and Literature Review

In this chapter, we provide technical background of this project as well as works that are relevant to our work. First we review the task of neural machine translation, including recent advances in prompting multilingual models for machine translation in Section 2.1. Then in Section 2.2 we review some of the decoding methods that are relevant to our work, together with the advantages and disadvantages of each method. And finally, we examine methods on optimizing language models to learn human preferences in Section 2.3.

2.1 Neural Machine Translation with Multilingual Models

2.1.1 Neural Machine Translation (NMT)

Machine Translation (MT) is a subfield of NLP aiming to develop automatic systems that translate texts from one language to another. More formally, given a source sentence $\mathbf{x}$ in language L_A, the task of machine translation is to generate a sentence $\mathbf{y}$ in the target language L_B that preserves the full semantic meaning of the source sentence. Neural Machine Translation (NMT) has rapidly become the dominant paradigm in MT over the past decade. Unlike previous Statistical Machine Translation (SMT) techniques, NMT deploys a single deep learning model that translate sentences in an end-to-end fashion [7, 3, 40]. The architecture for NMT typically follows the encoder-decoder networks [22]. First introduced by Kalchbrenner and Blunsom [16], the encoder maps the source sentence to a fixed-length vector representation that captures the information needed for translation, and the decoder generates the target sentence conditioned on the representation of the source sentence. The use of the fixed-length representation, however, results in low-quality translations for long source sentences [7]. Bahdanau et al. [3] proposed to use *attention* to replace the fixed-length representation. The attention mechanism allows the decoder to utilize the most relevant parts

in the source sentence for generating the current token in the output. For a comprehensive review on the development of NMT, we refer to the survey paper by Stahlberg [38].

Although NMT has been gaining tremendous attention from NLP researchers, it remains an active research area with many unsolved issues. One of the main challenges for developing NMT models is the heavy reliance on abundant parallel texts. To train these models, the dataset contains a set of source sentences, and another set of reference translations for the source sentences. However, it is expensive and time-consuming to construct this type of training data, because it requires human translators to translate each source sentence. And it becomes more difficult for low-resource languages. Back-translation [33, 35] is a common approach to mitigate this issue by automatically generating parallel data based on monolingual texts. Another issue with previous NMT system is the lack of generalizability. A system trained for a specific language pair (for example from Chinese to English) may perform poorly when confronted with a different language pair (like German to English). Therefore, to build a robust translation system, multiple datasets consisting of parallel texts in various domains and language pairs are needed.

Recently, works on large multilingual models have shown potential for mitigating the aforementioned issues. In the next two sections, we review these newly proposed models and approaches to use these models for machine translation.

2.1.2 Multilingual Large Language Models (MLLMs)

The advances in Large Language Models (LLMs) [6, 8, 42, 43] have led the rapid development of Multilingual Large Language Models (MLLMs). While LLMs like GPT-3 [6] and PaLM [8] are trained on large amount of texts mostly in English, MLLMs are trained on multilingual datasets that consist of monolingual texts in multiple languages. This endows MLLMs with the ability to perform NLP tasks in multiple languages, as well as cross-lingual tasks like translation, with a shared vocabulary and set of parameters.

Many MLLMs were developed from existing LLMs, essentially creating multilingual versions of the original models. For example, mBERT [10] and mT5 [50] are extensions of the BERT [10] and T5 [29] models respectively. Therefore, MLLMs, like their LLMs counterpart, come with different architectures:

- **Encoder-decoder** is the original architecture used in Transformer [44]. This architecture is suitable for sequence-to-sequence tasks and is adopted in mT5 [50].
- **Encoder-only** architecture only utilizes the encoder module in the Transformer architecture. It is mainly designed to make prediction on the token level in the input sentence, using the full context of the sentence. mBERT [10] is an example.

- **Decoder-only** architecture only utilizes the decoder module in Transformer, and is used for causal language models like XGLM [19] to generate sequences.

A typical way to pre-train an MLLM is to inherit the pre-training pipeline of an existing LLM. For example, mT5 closely follows the training procedure of T5 but on the mC4 dataset, which is a multilingual extension of the original C4 dataset including 101 languages [50].

Of particular relevance to this project, decoder-only MLLMs have demonstrated strong language modelling and few-shot learning capabilities inherited from monolingual LLMs. They typically follow the similar causal language modelling objective for pre-training monolingual causal language models. XGLM [19] uses the decoder-only model architecture similar to that of GPT-3 [6], and was pre-trained on CC100-XL that covers 68 Common Crawl (CC) snapshots and 134 languages. Experiment results show that XGLM (7.5B) outperforms GPT-3 (6.7B) on multilingual commonsense reasoning, natural language inference, and translation tasks. BLOOM significantly increases the scale of pre-training and model size up to 176 billion parameters.

Beyond pre-trained MLLMs, BLOOMZ [21] was fine-tuned on BLOOM using the multitask fine-tuning strategy [46, 31], with the goal of enhancing the zero-shot inference ability of pre-trained MLLMs. Muennighoff et al. [21] extended the P3 dataset [31] consisting of supervised data for multiple NLP tasks, and created the xP3 dataset that includes additional multilingual tasks with both English and non-English prompts. They found that, after fine-tuning BLOOM on xP3, the resulting BLOOMZ model achieved impressive zero-shot results on English and non-English tasks.

Since MLLMs are multilingual extensions of monolingual LLMs, they retain the capability of zero- and few-shot inference [19]. With only a few or even no demonstration examples, MLLMs can perform tasks like translation by conditioning on the input prompt. This ability builds the basis of in-context few-shot learning for NMT using MLLMs.

2.1.3 Prompting MLLMs for NMT

Unlike previous encoder-decoder NMT models that learn how to map source sentences to their translations, MLLMs are pre-trained to model multiple languages with a shared lexicon. Therefore, with a proper input prompt, the strong language modelling ability enables MLLMs to directly perform translation without the need for large amount of parallel texts [12, 19, 53, 45, 54, 1].

The general pipeline of this new approach, prompting MLLMs for NMT, follows the in-context few-shot learning framework. Given a language model with parameter θ, we construct a prompt that contains a few demonstration examples $(\mathbf{x}^{(1)}, \mathbf{y}^{(1)})...(\mathbf{x}^{(m)}, \mathbf{y}^{(m)})$

where $\mathbf{x}^{(\cdot)}$ is the source sentence written in the source language and $\mathbf{y}^{(\cdot)}$ is the corresponding translation written in the target language. Added to this is the query sentence $\mathbf{x}_q$ that we want the language model to translate. The combined input that will be passed to the model is:

$$\mathbf{x} = T(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}, ..., \mathbf{x}^{(m)}, \mathbf{y}^{(m)}, \mathbf{x}_q) \tag{2.1}$$

where $T(\cdot)$ is the prompt template. The model is then tasked with completing the prompt in an auto-regressive way. This means that at time step t, the model generates a probability $p(y_t|y_{<t}, \mathbf{x}, \theta)$ contributing to form a completed output sentence $\mathbf{y}$ (we will discuss this in Section 2.2), which is the translation we asked for.

The efficacy of this method has been explored by several recent works. Garcia et al. [12] trained bi-lingual and tri-lingual language models, assessing their in-context few-shot learning ability on the WMT'21 new translation task [2]. They showed that the performance of these self-supervised models could rival specialized translation systems with five demonstration examples. They, along with Vilar et al. [45] and Zhang et al. [53], also found that the quality of the examples plays a crucial role in the quality of the output translation. They noted that certain attributes (formality and regional varieties) of the examples can be used to control the style of the translation.

Further investigations have been made into the selection of demonstration examples. Zhang et al. [53] noted that the semantic similarity and LLM likelihood, among other features, "show significant Spearman correlation with their prompting performance; yet, none of the correlations are strong enough". They also identified several issues with the in-context few-shot learning pipeline, such as off-target translation and prompt trap. Agrawal et al. [1] emphasized the importance of example domains for translation performance and proposed an example selection strategy based on n-gram overlap.

In summary, prompting MLLMs for in-context few-shot learning is a new paradigm towards the task of NMT. Compared to traditional specialized translation systems, this method offers greater generality and flexibility, and significantly reduces the demand for human labor.

2.2 Decoding Methods

The decoding process is an essential step for Natural Language Generation (NLG) using LLMs. As discussed previously, at each time step, the model generates a probability score for each token in the vocabulary. In this section, we review some methods, grouped by objectives, that utilize these scores to generate natural language sentences.

2.2.1 MAP Decoding

The maximum-a-posteriori (MAP) decoding aims to select the most likely output sentence, defined by:

$$\mathbf{y} = \arg\max_{\mathbf{y}\in\Sigma^*} P(\mathbf{y}|\mathbf{x},\theta) = \arg\max_{\mathbf{y}\in\Sigma^*} \prod_{t=1}^{T} P(y_t|y_{<t},\mathbf{x},\theta) \tag{2.2}$$

However, search for the optimal $\mathbf{y}$ may potentially involve traversing all possible sentences, which is not possible in practice. Several methods have been proposed to approximate the optimal output, among which greedy decoding and beam search are widely used.

Greedy decoding is the most straightforward decoding method. As the name suggests, greedy decoding selects the token associated with the highest predicted probability at each time step. Formally, at time step t, we select the token as

$$y_t = \arg\max_{y\in\Sigma} P(y|y_{<t},\mathbf{x},\theta) \tag{2.3}$$

Despite being simple and fast, it is unlikely that greedy decoding meets the MAP criterion.

Beam search extends beyond greedy decoding by maintaining multiple hypotheses. The number of hypotheses is denoted as N, and is often referred to as the beam width. At the beginning of time step t, beam search algorithm keeps N partially generated hypotheses with length $< t$: $\{\mathbf{y}^k_{<t}\}^N_{k=1}$. Then the algorithm expands each hypothesis by adding N most likely next token. Hypotheses that have already ended with the $< eos >$ token are not expanded. The algorithm selects N most likely partial hypotheses out of the $\sim N^2$ proposals, resulting in N hypotheses with length $< t+1$ and moves on to time step $t+1$.

By considering multiple alternative output prefixes, beam search is able to find more globally higher likelihood sequences . It is a more generalized approach[1] that yields hypotheses with higher model scores. However, it is still an approximation to the MAP objective and outputs from beam search often lack of semantic diversity. Stahlberg and Byrne [39] showed that more exhaustive searching might not yield good translations, and shorter outputs or even empty output can be preferred by beam search. Still, it balances quality and efficiency, thus being a popular choice for machine translation.

2.2.2 Sampling-based Decoding

In contrast to MAP decoding methods that produce deterministic results, sampling-based methods introduce randomness into the decoding process and select each token by sampling

[1]When $N = 1$, beam search degenerates to greedy decoding.

from the probability distribution over the vocabulary. At time step t, the token is selected as

$$y_t \sim P(y|y_{<t}, \mathbf{x}, \theta) \tag{2.4}$$

This basic sampling method generally produces more diverse outputs compared to MAP decoding, but due to its stochastic nature, it may produce low-quality outputs. More advanced sampling methods are proposed to modify the probability distribution from which the next token is sampled.

Temperature sampling introduces a temperature parameter τ that controls the smoothness of the distribution. The new probability for a token in the vocabulary is re-normalized as

$$\begin{aligned} \tilde{p}(y_t|y_{<t}, \mathbf{x}, \theta) &= \frac{\exp(l_\theta(y_t|y_{<t}, \mathbf{x})/\tau)}{\sum_{y\in\Sigma}\exp(l_\theta(y|y_{<t}, \mathbf{x})/\tau)} \\ &= \frac{p(y_t|y_{<t}, \mathbf{x}, \theta)^{\frac{1}{\tau}}}{\sum_{y\in\Sigma} p(y|y_{<t}, \mathbf{x}, \theta)^{\frac{1}{\tau}}} \end{aligned} \tag{2.5}$$

where $l_\theta(y_t|y_{<t}, \mathbf{x})$ is the logit of the token y_t from the model (the raw score for predicting the token y_t before the softmax layer). When τ is large, the new distribution is more similar to the original one. As we decrease the value of τ, the probabilities of the previously likely tokens increase and the output sentences are less diverse. When $\tau \to 0$, we only select the most probable token and temperature sampling becomes greedy decoding.

Top-k sampling is a simple approach that adjusts the vocabulary from which we sample tokens. We make a ranking of all the tokens in the original vocabulary, based on their probability scores. Then we select the top k tokens, where k is a hyper-parameter, re-normalize the probability distribution over these k tokens, and sample from this new distribution.

Top-p (nucleus) sampling [13] is similar to top-k sampling. Instead of picking the top k tokens, nucleus sampling first picks the least amount of tokens whose summed probability mass occupies a certain amount p. Formally, we want the smallest subset of the original vocabulary $\Sigma' \subseteq \Sigma$ that satisfies

$$\sum_{y\in\Sigma'} p(y|y_{<t}, \mathbf{x}, \theta) \geq p \tag{2.6}$$

We then sample from the re-normalized distribution over the new vocabulary Σ'.

Beam sampling combines elements of both beam search and sampling methods. As with beam search, beam sampling keeps track of N hypotheses. However, instead of expanding each hypothesis by adding N most likely tokens, beam sampling expands each hypothesis

by sampling from the token probability distribution. This introduces randomness into the beam search process while still retaining high likelihood outputs. Therefore, beam sampling provides a hybrid approach between deterministic and stochastic decoding.

2.2.3 Minimum Bayes Risk (MBR) Decoding

Both MAP decoding and sampling methods have certain limitations. MAP decoding focuses on likelihood of the outputs at the cost of diversity and length bias, while sampling methods promote diversity but lack of control over the outputs. Minimum Bayes Risk (MBR) decoding [18] provides an alternative objective for language generation [41].

MBR aims to directly minimize the expected loss between model outputs and references. First we define H as the hypothesis set, E the evidence (or reference) set, and $L(\cdot,\cdot)$ a loss function between two sentences. Next we define the risk of a hypothesis $\mathbf{y} \in H$ as

$$R(\mathbf{y}|\mathbf{x},\theta) = \mathbb{E}_{\mathbf{y}' \sim p_(\cdot|\mathbf{x})}[L(\mathbf{y},\mathbf{y}')] = \sum_{\mathbf{y}' \in E} L(\mathbf{y},\mathbf{y}')p(\mathbf{y}'|\mathbf{x}) \tag{2.7}$$

which measures the risk of choosing $\mathbf{y}$ as the final output. In practice, we do not have access to the true distribution of translations, so we approximate the references using outputs from the model itself. Since sentences in both H and E are generated from the model with parameter θ, we can set these two set to be the same $H = E$. MBR decoding then selects the hypothesis from H that has the lowest risk:

$$\mathbf{y}^* = \arg\min_{\mathbf{y} \in H} R(\mathbf{y}|\mathbf{x},\theta) = \arg\min_{\mathbf{y} \in H} \sum_{\mathbf{y}' \in H} L(\mathbf{y},\mathbf{y}')p(\mathbf{y}'|\mathbf{x},\theta) \tag{2.8}$$

The loss function can be any string-to-string distance function, or the negative of string-to-string alignment function such as BLEU [24] and BLEURT [34] (see Section 3.3.2 for details). Elements in H and E are generated by the model via either beam search or sampling. When they are generated by sampling, i.e., $H = \{\mathbf{y} \sim p(\mathbf{y}|\mathbf{x},\theta)\}$, we can omit the last term in Equation 2.8.

It can be seen that MBR is a form of consensus decoding. The hypothesis that is the most similar to other hypotheses is favored by MBR. Therefore, the number, diversity and quality of the hypotheses are important to the performance of MBR decoding, and we will analyze the effects of these factors in Section 4.1.2.

In summary, the outline of MBR decoding is the following:

1. Generate a set of hypotheses H from the model, using either beam search or sampling.
2. Select the hypothesis with the lowest risk as in Equation 2.8.

Although the algorithm is conceptually simple, the computational complexity of MBR is a concern. Step 1 might be efficient for small models, but it can be time-consuming when using MLLMs. In step 2, the risk computation scales quadratically with the number of hypotheses. Although MBR produces better outputs for translation tasks [41], we hypothesize that the time complexity of MBR decoding hinders its practical adoption. In this work, we aim to teach the model to learn the behavior of MBR decoding, thereby combining the high performance of MBR with the computational efficiency of standard decoding methods.

2.3 Preference Optimization for LLMs

Despite the remarkable capabilities of LLMs, one of the main challenges remains to control the behavior of the LLMs. Given the vast amount and diversity of their pre-training data, they are likely to be exposed to harmful and misleading information and can inadvertently mirror both the helpful and harmful aspects of the training data and reflect them in the outputs [23, 4]. Therefore, techniques have emerged to fine-tune LLMs to better align with human preferences, aiming at advancing the safety, helpfulness, and controllability of the models. In this section, we review two strategies that target this preference learning: Reinforcement Learning from Human Feedback (RLHF) and Direct Preference Optimization (DPO).

2.3.1 Reinforcement Learning from Human Feedback (RLHF)

RLHF is one of the most widely used algorithms for learning human preference [9, 55]. Here we follow the review provided in [28] for consistent notations. Starting with a pre-trained LLM, RLHF follows a three-stage pipeline.

The first stage is supervised fine-tuning (SFT) where the LLM is fine-tuned on datasets for downstream tasks (for example, from BLOOM to BLOOMZ). We denote this supervised fine-tuned model as π_{SFT}, and we will also refer to it as the base model.

The second stage is to train a reward model $r_\phi(\mathbf{x},\mathbf{y})$, which predicts human preference over two responses $(\mathbf{y}_1,\mathbf{y}_2)$ generated from π_{SFT} with prompt $\mathbf{x}$. The dataset for training such a reward model is usually a preference dataset. Given the prompt $\mathbf{x}$ and the two responses, human labelers are asked to select the preferred response $\mathbf{y}_w$ (winner) and the dispreferred response $\mathbf{y}_l$ (loser), thereby creating a large set of preference tuples $(\mathbf{x},\mathbf{y}_w,\mathbf{y}_l)$. Suppose there is a ground-truth reward function $r^*(\mathbf{x},\mathbf{y})$ that expresses the true reward value, the distribution of human preference can be modelled by the Bradley-Terry (BT) model [5] as

$$p^*(\mathbf{y}_1 \succ \mathbf{y}_2|\mathbf{x}) = \frac{\exp(r^*(\mathbf{x},\mathbf{y}_1))}{\exp(r^*(\mathbf{x},\mathbf{y}_1)) + \exp(r^*(\mathbf{x},\mathbf{y}_2))} = \sigma(r^*(\mathbf{x},\mathbf{y}_1) - r^*(\mathbf{x},\mathbf{y}_2)) \tag{2.9}$$

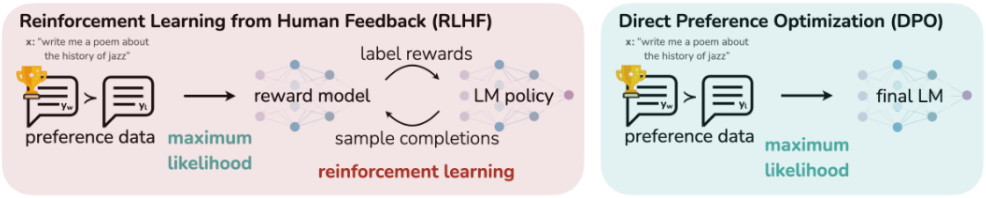


Fig. 2.1 The figure is from Rafailov et al. [28] and it demonstrates the pipelines of RLHF and DPO.

With the preference dataset $D = \{\mathbf{x}^{(i)}, \mathbf{y}_w^{(i)}, \mathbf{y}_l^{(i)}\}_{i=1}^N$, the reward model r_ϕ can be trained with the following objective:

$$L_R(r_\phi, D) = -\mathbb{E}_{(\mathbf{x},\mathbf{y}_w,\mathbf{y}_l)\sim D}[\log\sigma(r_\phi(\mathbf{x},\mathbf{y}_w) - r_\phi(\mathbf{x},\mathbf{y}_l))] \tag{2.10}$$

where $\sigma(\cdot)$ is the sigmoid function.

Based the trained reward model, the third stage is to teach the LLM to learn from the feedback provided by the reward model. To formulate, we have two models (policies), the reference policy π_{ref} and the language model policy π_θ. Both models are initialized from the SFT model π_{SFT} but the reference policy is frozen during training. The optimization problem of this stage is the following

$$\max_{\pi_\theta} \mathbb{E}_{\mathbf{x}\sim D,\mathbf{y}\sim\pi_\theta}[r_\phi(\mathbf{x},\mathbf{y})] - \beta\mathbb{D}_{\text{KL}}[\pi_\theta(\mathbf{y}|\mathbf{x})||\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})] \tag{2.11}$$

where the KL divergence term weight β controls how far the language model policy deviates from the reference policy. This problem is often solved with reinforcement learning algorithms like Proximal Policy Optimization (PPO) [32].

2.3.2 Direct Preference Optimization (DPO)

Although RLHF has been widely used and has shown great success, the whole procedure is complex. The PPO algorithm is sensitive to hyperparameters [51] and the RLHF training process suffers from high computational costs compared to supervised learning [28]. Rafailov et al. [28] proposed a algorithm called Direct Preference Optimization (DPO) that aims to sidestep the reward modelling step and optimize the model without the need for reinforcement learning, as seen in Figure 2.1. To do that, DPO approximates the reward function using the language model policy π_θ and the reference policy π_{ref}.

DPO starts with the same optimization problem in Equation 2.11 with a general reward function $r(\mathbf{x},\mathbf{y})$. It can be shown that the optimal policy that satisfies this objective can be

written as

$$\pi_r(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})\exp(\frac{1}{\beta}r(\mathbf{x},\mathbf{y})) \tag{2.12}$$

where $Z(\mathbf{x})$ is the partition function that is intractable in practice. With some reparameterization, we can represent the reward function in terms of its corresponding optimal policy as

$$r(\mathbf{x},\mathbf{y}) = \beta\log\frac{\pi_r(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})} + \beta\log Z(\mathbf{x}) \tag{2.13}$$

This is an essential step in DPO as we can now express the human preference distribution using only the policies, when r is the ground-truth reward function r^*, as

$$p^*(\mathbf{y}_1 \succ \mathbf{y}_2|\mathbf{x}) = \sigma(\beta\log\frac{\pi_{r^*}(\mathbf{y}_1|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_1|\mathbf{x})} - \beta\log\frac{\pi_{r^*}(\mathbf{y}_2|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_2|\mathbf{x})}) \tag{2.14}$$

The change of variable from Equation 2.9 to Equation 2.14 let us straightforwardly formulate a maximum likelihood objective for training π_θ that approximates π_{r^*}:

$$L_{\text{DPO}} = -\mathbb{E}_{(\mathbf{x},\mathbf{y}_w,\mathbf{y}_l)\sim D}[\log\sigma(\beta\log\frac{\pi_\theta(\mathbf{y}_w|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_w|\mathbf{x})} - \beta\log\frac{\pi_\theta(\mathbf{y}_l|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_l|\mathbf{x})})] \tag{2.15}$$

DPO, in comparison to RLHF, offers a more straightforward algorithm while sharing the same objective of preference learning. Therefore, the DPO algorithm is well suited for our task of aligning the model with MBR decoding preferences, though in our case, the reward signals are obtained from automatic MBR decoding process.

Other concurrent works such as Preference Ranking Optimization (PRO) [37] also focus on learning preference without the need for reinforcement learning. The pair-wise PRO loss can be seen as a special case of DPO loss. In DPO, the reward function is represented with both the language model policy and the reference policy, which is usually initialized from the same base language model. However, the PRO objective, in pair-wise setting, approximates the reward function with a reference policy that assigns equal probability to all responses.

2.4 Summary

In this chapter, we have reviewed technical background on multilingual large language models for NMT, decoding methods, and preference optimization techniques to motivate the contributions of this project. We discussed recent advances in prompting MLLMs for in-context few-shot NMT, setting the basis for our exploration into the effectiveness of this approach. We reviewed several decoding methods and identified MBR decoding as a promising method whose limitations we aim to address. Finally, we described recent

preference optimization techniques that inspire our proposed fine-tuning approach to align MLLMs with MBR decoding. In the next chapter, we introduce our methods for exploring in-context NMT and self-supervised preference optimization for MLLMs.

Chapter 3

Methodology and Experiment Design

In this chapter, we introduce our proposed methods for studying NMT with MLLMs. As mentioned previously, this project consists of two main parts. For the first part, we present our strategy to select demonstration examples, the prompt format, and detailed configurations of the decoding methods we use. For the second part, we describe our preference optimization algorithm, including our schema for constructing the preference dataset, and the training objective we use to fine-tune the models.

3.1 In-context Learning for NMT

Our goal for the first part is to gain insights into prompting MLLMs for NMT within the in-context learning framework. We aim to unravel the answer to our first research question on the potential performance gain through decoding methods. We implement the in-context learning pipeline following previous work [12, 53]. Specifically, we construct an input prompt consisting of a few translation pairs as demonstration examples and the query sentence, according to Equation 2.1. The prompt is then fed into the language model, and a decoding method is applied to form a translation based on the model outputs. We investigate how key factors like example selection and decoding methods impact translation performance. Next, we elaborate on each of the factors experimented in this project.

3.1.1 Selection of Demonstration Examples

The selection of demonstration examples is a key component to in-context learning. These examples serve two main purposes: providing the model with enough context of the task and evoking the knowledge that the model has acquired during its pre-training phase.

There are three main design choices with selecting examples: (1) the number of demonstration examples, (2) the underlying dataset from which we draw the examples, and (3) the exact strategy to select examples from the dataset given the query sentence. Below we go through each one of the design choices.

The number of demonstration examples has a strong correlation with the model's performance [53]. However, there is a trade-off: increasing the number of demonstration examples means increasing the input length, which causes burden to both computation time and memory consumption. Garcia et al. [12] used 5 examples and showed impressive results on various WMT datasets. In this project, we test the impact of using 2 to 32 demonstration examples.

The selection pool, which is the dataset from which we draw our examples, inherently determines several features of the demonstration examples, including the source-target language pair, quality, and the domain. Following previous works [12], our chosen selection pool is the validation set from the WMT news translation task. Our rationale is twofold: First, the WMT validation set is created by expert human translator, and previous work [45] have attested the superior translation quality of the validation set. Second, the validation set shares the same domain as the test set, which allows us to focus on in-domain translation.

The selection strategy has been explored by previous works [1, 54]. Zhu et al. [54] found that randomly selecting examples from a high-quality dataset gives a strong baseline. Building on this insight, we adopt random example selection for most of the experiments in the first part of the project. To elaborate, for each query sentence in the test set, we randomly select a fixed number of examples from the validation set without replacement. As a comparison, we also explore selecting examples based on semantic similarity between the query and the examples. As for the second part of the project, we randomly select 8 examples from the validation set, and utilizing these 8 examples to construct prompts for all the test sentences (we will discuss the motivation for that in Section 3.2).

3.1.2 Prompt Construction

LLMs and MLLMs can be very sensitive to the prompt format, as previous studies have shown. Although exploring handcrafted prompt formats offers insights, it is a tedious process. The complexity is magnified for MLLMs due to the potential of employing cross-lingual prompts. In this project, we adopt a straightforward prompt format that has been shown successful in previous works. Formally, let $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$ be the source sentence and target translation sentence respectively from the ith demonstration example, and $\mathbf{x}$ be the query

sentence we want the model to translate. Then the simple prompt is constructed as

$$\mathbf{x}^{(1)} = \mathbf{y}^{(1)}$$
$$\ldots$$
$$\mathbf{x}^{(m)} = \mathbf{y}^{(m)}$$
$$\mathbf{x} =$$

Note that though there is no language tag or language name in the prompt, and the model is able to infer the target language from the examples. While future work could investigate further on prompt tuning for MLLMs, sticking with this simple prompt format allows us to focus on the effects of other factors like example selection and decoding methods. Other prompt formats that we attempted are listed in Appendix A for reference.

3.1.3 Decoding Methods

Decoding strategy significantly impacts the quality of in-context NMT. While most works on NLG and NMT focus on standard decoding methods like greedy decoding and beam search, Garcia et al. [12] and Suzgun et al. [41] have shown the efficacy of MBR decoding on NMT. In this project, we systematically investigate various decoding methods to optimize translation performance.

Beam search typically benefits from a larger beam width, as it expands the search space thus enhancing the probability of finding outputs with globally higher likelihood. We investigate the effect of beam width on the translation task by varying the beam width from 1 (greedy decoding) to 8. This range was chosen based on compute and memory constraints. Our experiments on beam search aim to test the validity of the MAP objective and build baselines for exploring other decoding methods.

MBR decoding that we described in Section 2.2 is a general outline. Here we list the key components of MBR that we will explore in this project.

1. Hypothesis generation: Our exploration predominantly leverages sampling-based decoding methods to generate hypotheses and form the set H. We assess the performance of temperature sampling, its combination with nucleus sampling, and beam sampling.

2. Number of hypotheses: The size of H is an important factor to the performance when using MBR decoding. By varying $|H|$ from 8 to 64, we assess it impact on performance, hypothesizing that larger $|H|$ can potentially lead to more accurate calculation of the Bayesian risk which leads to superior-quality output and more accurate ranking of the hypotheses.

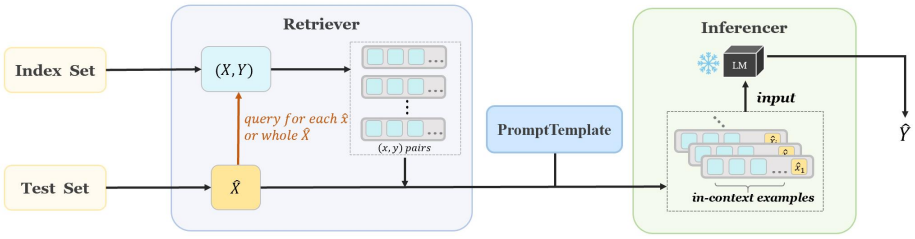


Fig. 3.1 The pipeline for the OpenICL framework for in-context learning. The figure is from Wu et al. [48]

3. Loss function: We choose the negative string-to-string alignment function as the loss function. Specifically, we use BLEURT [34], a learning-based metric, as our alignment function, which simultaneously serves as our evaluation metric. In this way, the MBR decoding process directly optimizes quality assessed by the evaluation metric. Despite the success of other alignment functions like BERTScore and Longest Common Subsequence (LCS) demonstrated in [41], we limit our exploration to BLEURT due to time constraints.

3.1.4 Post-processing

Prompting MLLMs for NMT can raise the issue of over-generation. In our case, the model might keep generating prompt-like outputs after the actual translation of the query finishes, by adding self-synthesized translation pairs. We formulate a systematic post-processing strategy based on the prompt format. For our simple prompt format, we simply truncate the output by removing tokens after the first new-line token generated by the MLLM.

3.1.5 Experiment Details

For the experiments on this part, in-context few-shot learning for NMT, we use two decoder-only MLLMs. The first one is XGLM [19] with 7.5 billion parameters. The model supports 30 languages covering 16 language families, ranging from high- and low-resource languages. The model architecture follows GPT-3 [6], containing 32 hidden layers with hidden dimension of 4096 and a much larger embedding layer than GPT-3 to support a joint vocabulary for all languages. The second model we explore is BLOOM with 7.1 billion parameters. This model supports 46 natural languages and 13 programming languages. The model architecture is modified from Megatron-LM [36] and has 30 layers with hidden dimension of 4096. Both

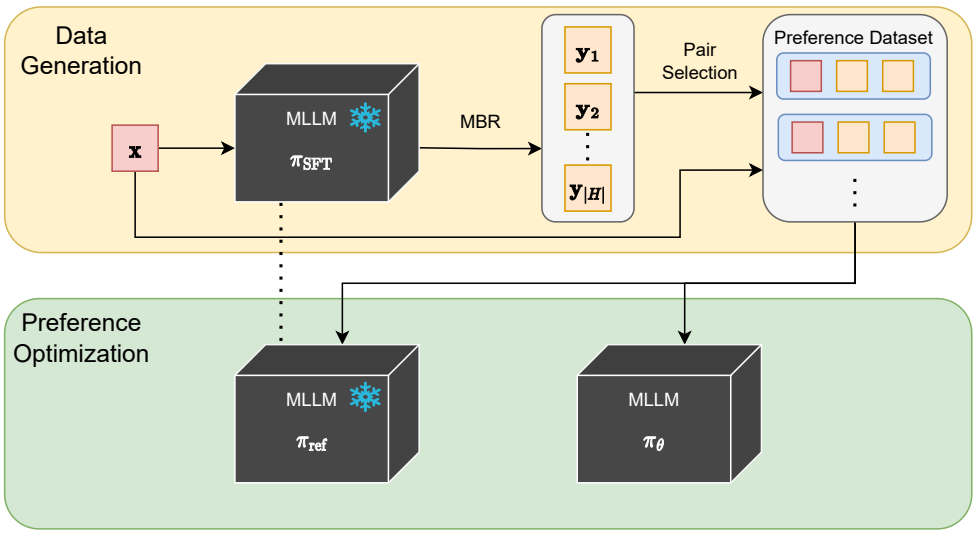


Fig. 3.2 The outline of our unsupervised preference optimization with MBR. The diagram only shows the data generation process for one source sentence $\mathbf{x}$. Both π_θ and π_{ref} are initialized from π_{SFT} and π_{ref} is frozen during training. The dotted line indicates that the two models are the same.

models have only been pre-trained and are not fine-tuned on any labeled dataset or parallel text.

We use the OpenICL framework [48] for in-context learning. The pipeline of the OpenICL inference process is shown in Figure 3.1. To select the demonstration examples based on semantic similarity (kNN selection [20]), we use the SentenceTransformer [30] library to extract the embeddings of the source sentences and use the FAISS library [15] to search for similar source sentences in the candidate pool. We follow [54] to use the *all-mpnet-base-v2* feature extractor and the similarity is calculated using cosine similarity.

All the experiments for this part were run on a single Nvidia A100 GPU.

3.2 Unsupervised Preference Optimization with MBR

In-context few-shot learning is an efficient and effective way to instruct MLLMs to do tasks that it has never been trained for, without the need for fine-tuning. However, in order to get good performance on the downstream task, inference time is sometimes sacrificed to handle very long input sequence (more demonstration examples) and to perform complicated decoding such as MBR. In the second part of the project, we propose a novel unsupervised method that utilizes the recently proposed Direct Preference Optimization (DPO) [28] and aligns the model with the behavior of MBR decoding. Specifically, we teach the language model to learn the preference ranking of the hypotheses given by MBR, and generalize this

preference learning ability to translation ability by assigning more probability to hypotheses with lower risks.

The training consists of two steps: (1) data generation and (2) preference learning. During the first step, we freeze the MLLM and feed it with the source sentences in the training dataset. For each source sentence, we ask the model to generate translation hypotheses using the MBR decoding procedure. This can be done in either few-shot or zero-shot way. Once we have the hypothesis set for each source sentence, and their associated risks, we construct the preference dataset by selecting pairs of hypotheses as ranked by MBR. The second step is to fine-tune the MLLM on the preference dataset for preference learning. The pipeline for this method is outlined in Figure 3.2. In the following sections, we provide details on each of the steps.

3.2.1 Data Generation

To start with, we describe our strategies to construct the preference dataset P. The preference dataset contains a list of triplets $(\mathbf{x}, \mathbf{y}_w, \mathbf{y}_l)$ where $\mathbf{x}$ is the input prompt containing the source sentence, $\mathbf{y}_w$ is the preferred (winning) hypothesis, and $\mathbf{y}_l$ is the dispreferred (losing) hypothesis. To generate this type of triplets, the pipeline consists of two main part.

The first step is to generate a set of hypotheses, i.e., the set H. This is similar to the first step of MBR decoding. For each source sentence in a monolingual dataset, we ask the model to generate a list of hypotheses, either through in-context learning or zero-shot inference on a model that has been fine-tuned for translation task. We use temperature sampling primarily with a temperature of 0.7. We also ask the model to generate one hypothesis using beam search with beam width of 4.

The second part is to select the winning hypothesis and the losing hypothesis to form the preference pair, which we call *pair selection*. Here we describe several pair selection strategies that we explore.

1. The simplest way to do pair selection for each source sentence $\mathbf{x}$ is to only select the hypothesis with lowest risk as the winner and the hypothesis with highest risk as the loser:

$$\mathbf{y}_w = \arg\min_{\mathbf{y} \in H} R(\mathbf{y}|\mathbf{x}, \theta),\ \mathbf{y}_l = \arg\max_{\mathbf{y} \in H} R(\mathbf{y}|\mathbf{x}, \theta) \tag{3.1}$$

   This method results in the largest risk difference between the winning and losing hypotheses. Since there is only a single pair per source sentence, the training process is the most efficient. We denote the resulting preference dataset as P_{single}.

2. We can include more pairs to the preference dataset to help the model learn the full ranking of hypotheses. First, we rank all the hypotheses according to their risks in ascending order: $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_{|H|}$. Then we select pairs by picking consecutive hypotheses in the ranked list as

$$(\mathbf{y}_1, \mathbf{y}_2), (\mathbf{y}_2, \mathbf{y}_3), ..., (\mathbf{y}_{|H|-1}, \mathbf{y}_{|H|}) \tag{3.2}$$

which results in $|H| - 1$ pairs per source sentence. However, neighboring hypotheses might be too similar to each other. To address this, we add stride to this pair selection strategy. For a stride of s, the selected pairs become:

$$(\mathbf{y}_1, \mathbf{y}_{1+s}), (\mathbf{y}_{1+s}, \mathbf{y}_{1+2s}), ... \tag{3.3}$$

until we hit the end of the ranked list. We denote the preference dataset with this type of pairs as P_{con}.

3. Instead of using only the outputs from temperature sampling, we can also include the output from beam search, denoted as $\mathbf{y}_{\text{beam}}$. Then we can construct three pairs $(\mathbf{y}_w, \mathbf{y}_{\text{beam}})$, $(\mathbf{y}_{\text{beam}}, \mathbf{y}_l)$, and $(\mathbf{y}_w, \mathbf{y}_l)$ where $\mathbf{y}_w$ and $\mathbf{y}_l$ are as defined in Equation 3.1. If the beam search output happens to have the lowest or highest risk, we remove the duplicated pairs and the pairs with same hypothesis. This method allows the model to contrast the best and worst outputs from MBR with the beam search output. Essentially, we use the beam search output as an anchor to teach the model to learn MBR preference. Preference dataset that are constructed using this strategy is denoted as P_{beam}.

4. It is also possible to include all possible pairs where the winning hypothesis has lower risk than the losing one. However, this results in $\frac{|H|(|H|-1)}{2}$ pairs per source sentence and is too costly to train. We denote this preference dataset as P_{all} and will only use it for analyzing MBR behavior in Section 4.1.3.

3.2.2 Training Objective

Once we have the preference dataset, the key part is to train the model to learn the preference. In this project, we explore two loss functions. The first one is the DPO loss in Equation 2.15 and the second one is the DPO loss combined with a language modelling loss.

The DPO loss can be directly applied to our preference dataset. We use BLOOMZ [21] as the SFT policy and use it to initialize the reference and language model policies. The underlying ground-truth reward function r^* is the risk from MBR decoding.

Training with DPO loss is an effective way for the model to learn preference. Although the model learns to prefer the winning translation over the losing one, the likelihood of the

better translation is not explicitly optimized. In fact, we found that training with the DPO loss leads to decreased likelihood of the winning translation compared to the reference policy. Therefore, we explicitly add a language modelling objective (negative log-likelihood loss) on $\mathbf{y}_w$ to the loss function. A simple way to do this is to combine the DPO and NLL losses together with a hyper-parameter γ that controls the contribution of the language modelling objective:

$$\begin{aligned} L &= L_{\text{DPO}}(\mathbf{x}, \mathbf{y}_w, \mathbf{y}_l) + \gamma L_{\text{NLL}}(\mathbf{x}, \mathbf{y}_w) \\ &= -\mathbb{E}_{(\mathbf{x},\mathbf{y}_w,\mathbf{y}_l)\sim P}[\log\sigma(\beta\log\frac{\pi_\theta(\mathbf{y}_w|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_w|\mathbf{x})} - \beta\log\frac{\pi_\theta(\mathbf{y}_l|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_l|\mathbf{x})}) + \gamma\log(\pi_\theta(\mathbf{y}_w|\mathbf{x}))] \end{aligned} \tag{3.4}$$

However, we would like a way to control the weight of NLL loss more precisely, down to each training instance. We propose a self-adaptive NLL loss based on the ratio of the likelihoods from the language model policy and the reference policy. Formally, the new language modelling objective is defined as

$$L_{\text{LM}} = \ln(1 + e^{w_\theta}), \text{ where } w_\theta = -\beta\log(\frac{\pi_\theta(\mathbf{y}_w|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_w|\mathbf{x})}) \tag{3.5}$$

When combined with the DPO loss, we have the second training objective:

$$L_{\text{ours}} = L_{\text{DPO}}(\mathbf{x}, \mathbf{y}_w, \mathbf{y}_l) + \gamma L_{\text{LM}}(\mathbf{x}, \mathbf{y}_w) \tag{3.6}$$

that we experiment with. The idea behind the design of this loss function is that, we want to put more weight on the language modelling objective when the language model policy assigns less probability to $\mathbf{y}_w$ than the reference policy, i.e., $\pi_\theta(\mathbf{y}_w|\mathbf{x}) < \pi_{\text{ref}}(\mathbf{y}_w|\mathbf{x})$. If we look at the gradient of the new language modelling objective:

$$\begin{aligned} \nabla_\theta L_{\text{LM}} &= \beta\frac{e^{w_\theta}}{1+e^{w_\theta}}\nabla_\theta L_{\text{NLL}} \\ &= \beta(1-\sigma(-w_\theta))\nabla_\theta L_{\text{NLL}} \\ &= \beta\sigma(w_\theta)\nabla_\theta L_{\text{NLL}} \end{aligned} \tag{3.7}$$

we can see that, w_θ controls the weight of the NLL loss, and is bounded by the sigmoid function. When $\pi_\theta(\mathbf{y}_w|\mathbf{x}) \ll \pi_{\text{ref}}(\mathbf{y}_w|\mathbf{x})$, w_θ will be a very large number and $\sigma(w_\theta)$ approximates 1. This means that when the language model policy fails to put more probability mass on the preferred hypothesis, the language modelling loss will enforce the model to learn it. Together with the DPO objective, the new loss function encourages the language model

policy to transfer probability mass from beam search output or the dispreferred output to the preferred output, which is selected via MBR decoding.

The main hyper-parameters that need to be tuned is γ which controls the weight of the language modelling objective, and β which controls how far away we want the language model policy to be from the reference policy. To see the effect of β, we can go back to the initial objective of preference optimization from Equation 2.11 (repeated here) where β is the weight for the KL divergence term.

$$\max_{\pi_\theta} \mathbb{E}_{\mathbf{x}\sim D,\mathbf{y}\sim\pi_\theta}[r_\phi(\mathbf{x},\mathbf{y})] - \beta\mathbb{D}_{\mathrm{KL}}[\pi_\theta(\mathbf{y}|\mathbf{x})||\pi_{\mathrm{ref}}(\mathbf{y}|\mathbf{x})]$$

3.2.3 Experiment Details

For the experiments on preference learning, the SFT model we use is BLOOMZ [21] with 7.1 billion parameters. This model was fine-tuned on BLOOM (7.1B) with multi-task instruction tuning as discussed in Section 2.1.2. We use this model as the reference policy and the starting point of the language model policy.

For the training set, we use the validation set from WMT'21 news translation task [2]. Note that we only use the source sentences from the validation set to self-synthesize the MBR hypotheses. In theory, any monolingual text dataset can be used as the training set and we leave to future work on potentially employing large monolingual datasets for training. Although the BLOOMZ model is capable of zero-shot inference as Muennighoff et al. [21] claimed, our initial trials do not show good performance. For consistency, we keep feeding the model with prompts containing 8 demonstration examples. In order to avoid data leakage, we randomly select 8 translation pairs from the validation set of WMT21, fix these 8 pairs as demonstration examples for all queries, and remove them from the training set. The 8 examples used for Chinese-English and German-English are shown in Appendix B.

For the preference dataset construction, we ask the BLOOMZ model to generate hypotheses with temperature sampling ($\tau = 0.7$) and beam search (beam width of 4). The alignment function used in MBR is BLEURT [34] with the BLEURT-20 [27] checkpoint.

For the training process, we train the model on 2 A100 GPUs for all experiments. We follow the recommendation from [28] to use the RMSprop optimizer and include a linear warm-up phase of 150 steps. We update all the parameters in the model and do not use the popular Low-Rank Adaptation method (LoRA) [14] because concurrent work [52] shows that LoRA is not effective for preference learning on translation task. We use the trainer provided by the official codebase of DPO[1].

[1] https://github.com/eric-mitchell/direct-preference-optimization/tree/main

	zh-en	en-zh	de-en
WMT'21	1948	1002	1000
WMT'20	2000	1418	785
WMT'19	2000	—	—
WMT'16	—	—	2999

Table 3.1 Number of translation pairs contained in various WMT test sets that we used in this project.

3.3 Evaluations

3.3.1 Datasets

We use the WMT dataset for all the experiments in this project. Following the suggestions from Garcia et al. [12] and Vilar et al. [45], we mainly focus on the test sets from recent years to avoid train/test overlap. Specifically, we use the test sets from WMT'21 news translation task [2] as our test sets with language pairs Chinese-English (zh-en), English-Chinese (en-zh), and German-English (de-en). We also use the WMT'16 German-English test set for some preliminary experiments. The validation sets associated with the test sets are the test sets from previous years. Specifically, we choose WMT'20 test sets as the validation sets for both training and example selection. In addition, we include the test set from WMT'19 Chinese-English as our held-out set to validate training performance. The detailed statistics of all the datasets we use are provided in Table 3.1 for reference.

3.3.2 Evaluation Metrics

Evaluation metrics for assessing the translation performance of translation systems can be roughly divided into three categories: (1) non-parametric metrics, (2) learned (parametric) metrics, and (3) human evaluation. In this project, we use two popular evaluation metrics from the first two categories respectively.

BLEU is widely used non-parametric for Statistical Machine Translation and Neural Machine Translation. It is easy to calculate and provides a basic intuition into the quality of the translation. BLEU is based on the combination of brevity penalty and n-gram overlap between the candidate translation (generated from the system) and the reference translation. Specifically, the n-gram overlap measures the precision of the translations and the brevity penalty serves as the recall term and penalizes candidates that are too short. We use the implementation from SacreBLEU [26] for both access to the WMT datasets and calculation of the BLEU score.

However, BLEU might not provide a robust evaluation of the translation quality due to its simplicity and non-parametric nature. It is overly sensitive to surface forms, thus punishing too hard on mistakes made by MLLMs as previous works [12, 45] have noted. Kocmi et al. [17] and Freitag et al. [11] have discouraged the use of BLEU to evaluate high-quality translations. Therefore, it might not be sufficient to use BLEU as the only metric to evaluate the quality of translations generated from MLLMs. We report BLEU scores in the next chapter mainly for completeness.

BLEURT [34] is a learned-based metrics based on deep learning models. It is based on BERT [10] and it has been trained to learn the semantic similarity between two candidates. Specifically, the BLEURT model is trained on the WMT Metrics Shared Task dataset, which contains human ratings for 430K sentence pairs. It is also trained on synthetic data, as an aid to learn from human evaluation. Although the original BLEURT metric was trained for English texts only, it has been extended to fit multilingual setting. We use the BLEURT-20 [27] checkpoint from the official GitHub repository[2] that supports non-English languages.

BLEURT has been shown to understand the semantic similarity between the candidate and the reference, which means that BLEURT is able to provide a more accurate and robust evaluation on translation quality. Therefore, in this project, we mainly focus on the BLEURT results to assess the performance of our models. Although BLEURT is trained to comopare two sentences in the same language, we found that BLEURT is capable of calculating cross-lingual similarity to some degree.

3.4 Summary

In summary, the method for this project is divided into two parts. For the first part, we follow previous works on prompting MLLMs for NMT and explore several factors in different modules of the pipeline that influence the performance of the model on translation task, with a focus on the performance of different decoding methods. And we described the implementation details of the in-context learning approach. For the second part, we propose a novel fine-tuning strategy that teaches the model to learn the preference of the MBR decoding method which allows effective and efficient inference with standard decoding methods. We specifically outlined our methods for constructing the preference dataset, and the training objective we used for preference learning and language modelling. We included the training details for reproducibility. Finally, we discussed the details of all the datasets we used for this project, and the evaluation metrics to assess the translation performance of the models.

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

Chapter 4

Results and Discussions

In this chapter, we present our key results and findings, offering both quantitative and qualitative analyses on the model performance. We first evaluate the effects of demonstration examples on in-context few-shot learning for NMT using MLLMs, setting up the basis for following experiments. Then we explore the performance of different decoding methods and identify MBR decoding as the promising one. Building on this insight, we investigate the behavior of MBR decoding, introducing the percentage of reverse pairs as our metric, and providing an in-depth case study on MBR outputs. Lastly, inspired by our early observations on MBR decoding, we experiment with our proposed fine-tuning method, demonstrating its efficacy.

4.1 In-context Learning for NMT

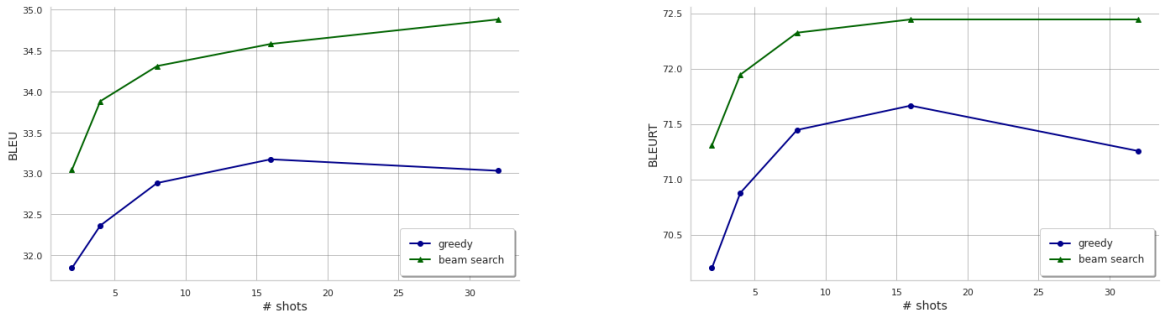
4.1.1 The Effect of Demonstration Examples

In our preliminary experiments, we evaluate the XGLM model on the test set of WMT16 German-to-English. To begin with, we randomly select translation pairs from the validation set of WMT16 German-English as demonstration examples, and vary the number of examples (number of shots) from a range of 2 to 32. The results, presented in Table 4.1, highlight a discernible trend: as we increase the number of examples, both BLEU and BLEURT scores increase, plateauing at 16 examples. And this pattern is consistent across both greedy decoding (beam width of 1) and beam search (beam width of 2).

The performance gains observed from 2-shot to 16-shot can be intuitively understood. As more demonstrations are provided, the model can more effectively elicit its pre-trained knowledge on the translation task. The added examples provide richer context, especially within the specific domain of news translation [49]. However, the diminished performance

Decoding	# Shots	BLEU↑	BLEURT↑
Greedy	2	31.84	70.2
	4	32.36	70.88
	8	32.88	71.45
	16	33.17	71.67
	32	33.03	71.26
Beam Search	2	33.04	71.31
	4	33.88	71.95
	8	34.31	72.33
	16	34.58	72.45
	32	34.88	72.45

Table 4.1 Performance comparison of different numbers of demonstration examples on the XGLM (7.5B) model. Beam search is conducted with beam width of 2.



(a) BLEU (b) BLEURT

Fig. 4.1 BLEU (a) and BLEURT (b) scores as functions of the number of the demonstration examples (shots). The figures contain the same results as in Table 4.1.

as the number of shots increases from 16 to 32 is more challenging to interpret holistically. One plausible explanation is the introduction of noise through the transitions between demonstration examples to the inference process[1], outweighing the benefit derived from these examples.

Apart from the number of demonstration examples, we investigate the effect of similarity between the examples and the query sentence. We adopt the BLOOM (7.1B) model on the test set of WMT21 Chinese-English, and retrieve 8 demonstration examples from the validation set of WMT21 Chinese-English. The example selection methods we employ are: (1) random selection and (2) *k*NN selection based on semantic similarity. For a more comprehensive analysis, we experiment with three decoding methods:

[1] https://ai.stanford.edu/blog/understanding-incontext/

Decoding	Selection	BLEU↑	BLEURT↑
Beam search	Random	19	62.19
	*k*NN	19.12	62.36
MBR w/ temp.	Random	12.91	61.27
	*k*NN	13.53	61.84
MBR w/ beam.	Random	18.83	62.94
	*k*NN	19.07	63.09

Table 4.2 Performance of BLOOM on WMT21 zh-en. Three decoding methods are experimented: beam search, MBR with temperature sampling, and MBR with beam sampling. For each decoding method, we apply random example selection and *k*NN example selection. *k*NN selection is effective but does not significantly improve translation performance. All the results are obtained after post-processing.

1. Beam search with a beam width of 4.
2. MBR using BLEURT as the alignment function, 8 hypotheses generated with temperature sampling $\tau = 0.9$.
3. MBR using BLEURT as the alignment function, 8 hypotheses generated with beam sampling, employing a beam width of 4 and temperature $\tau = 0.9$.

The resulting BLEU and BLEURT scores are shown in Table 4.2

Upon examining the results across each decoding method, we can see that the *k*NN selection strategy consistently improves performance measured by both BLEU and BLEURT. Nevertheless, the improvement is not significant. This finding aligns with the results presented in other works [53]. A potential reason for this slight improvement is that the 8 in-domain demonstration examples already sufficiently capture the task's essence, namely the input and output distributions and the input-output mapping as discussed in [49]. If the candidate pool from which we draw the examples is a more general dataset encompassing multiple domains, it is conceivable that example selection based on semantic similarity can yield more pronounced performance gain.

In summarizing our exploration of demonstration examples, it is clear that increasing the number of such examples is an effective way to enhance the translation performance of the model. However, considering the constraints of time and compute resources, we choose to use 8 examples for the majority of our subsequent experiments. This decision is informed by the observation that performance improvements tend to diminish with more than 8 examples. We will stick to the efficient random example selection strategy in our following in-context learning experiments.

# Shots	Beam width	BLEU	BP	BLEURT
4	1	32.36	0.99	70.88
	2	33.88	0.96	71.95
	4	34.19	0.94	72.22
16	1	33.17	0.98	71.67
	2	34.58	0.96	72.45
	4	34.76	0.94	72.6

Table 4.3 Performance of beam search with different values of beam width, evaluated with 4 and 16 demonstration examples in the prompt. BP is the brevity penalty term used in the calculation of BLEU where larger BP indicates longer sequence.

4.1.2 The Performance of Decoding Methods

In order to systematically analyze this phenomenon, we conduct a series of experiments employing varied configurations of decoding methods.

Beam Search

We initiate our investigation using the XGLM model on the WMT16 German-English dataset to examine the effect of beam width in beam search. This essentially serves to assess the efficacy of the MAP objective illustrated in Equation 2.2, which finds the output sentence with maximum likelihood. Specifically, we vary the beam width from 1 (equivalent to greedy decoding) to 4 with 4 and 16 examples respectively, and the results are reported in Table 4.3. We can see improvement in both BLEU and BLEURT as we increase the beam width, which indicates that translation with larger likelihood is also favored by the evaluation metrics. However, the marginal gain in performance diminishes when increasing the beam width from 2 to 4, compared to the gain fro 1 to 2. This plateau could be attributed to beam search not able to find considerably more probable outputs beyond beam width of 2, or that more likely outputs do not necessarily equate to better translations as assessed by BLEU and BLEURT [39]. The latter can be seen from the brevity penalty term in the calculation of BLEU score. As we increase the beam width, the brevity penalty exhibits a consistent decline, indicating shorter outputs. This decline negatively affects the BLEU score and it is possible that it also has a negative effect on BLEURT score, though implicitly.

MBR Decoding

We then direct our attention to MBR decoding. We evaluate the BLOOM model on three datasets: WMT21 Chinese-English, WMT21 English-Chinese, and WMT16 German-English

Dataset	Decoding	BLEU	BLEURT
WMT21 zh-en	Beam search	**19**	62.19
	MBR w/ beam.	18.83	**62.94**
	MBR w/ temp.	12.91	61.27
WMT21 en-zh	Beam search	26.46	61.11
	MBR w/ beam.	**26.65**	**62.52**
	MBR w/ temp.	18.88	62.36
WMT16 de-en	Beam search	27.38	65.16
	MBR w/ beam.	**27.44**	**65.65**
	MBR w/ temp.	19.9	65.49

Table 4.4 Performance of the three decoding methods (1) beam search, (2) MBR with beam sampling, and (3) MBR with temperature sampling, on three datasets. All results are obtained after post-processing.

with three decoding methods. Specifically, for (1) beam search which serves as the baseline, we set beam width to be 4, and length penalty to be 0.1, for (2) MBR with beam sampling, we use 8 hypotheses, and set beam width at 4 and length penalty at 0.1 and temperature at 0.9, and for (3) MBR with temperature sampling, we use 8 hypotheses generated with temperature of 0.9. Results, all after post-processing, are reported in Table 4.4. It is evident that MBR with beam sampling almost consistently outperforms the other methods. Notably, MBR with temperature sampling delivers good results on WMT21 English-Chinese and WMT16 German-English in terms of BLEURT score, surpassing the beam search baseline.

To gain deeper insights into MBR decoding when using the two sampling methods, we choose two representative examples from the WMT21 Chinese-English test set. The hypotheses generated as well as the final MBR outputs are provided in Table 4.5. Upon analysis, there is an obvious difference between the two sampling methods: beam sampling tends to generate multiple outputs that are very similar, with some even being identical. On the other hand, temperature sampling offers greater diversity in hypotheses, though it should noted that the quality of many hypotheses is less than optimal.

Specifically, for the first example, two methods result in the same final output. However, this agreed translation is not exactly accurate as "help" is not the same as "rescue". We can see that beam sampling generates this output multiple times but temperature sampling only generates this output once, among other very different translations including the one with the highest BLEURT score with the reference: *One rescue helicopter arrived to carry out rescue*.

Source	一架救援直升机抵达进行援救。
Reference	A rescue helicopter arrived to conduct rescue operations.
Beam sampling	1. An emergency helicopter arrived to provide rescue. 2. An emergency helicopter arrived to help. 3. An emergency helicopter arrived to help. 4. An emergency helicopter arrived to help.
Temperature sampling	1. An emergency helicopter arrived to help. 2. One rescue helicopter arrived to carry out rescue. 3. An AH-1Z Super Cobra helicopter landed to provide aid. 4. An emergency helicopter arrived to assist.
Source	乡村振兴，人才是关键。
Reference	Talent is the key to rural revitalization.
Beam sampling	1. To promote rural revitalization, talent is key. 2. To promote rural revitalization, talent is key. 3. Rural revitalization, talent is key. 4. Rural revitalization, talent is key. 5. The rural revitalization program is key. 6. To promote rural revitalization, talent is key. 7. Rural revitalization, people are key. 8. The Rural Revitalization, People are Key.
Temperature sampling	1. Successful transformation from rural backwardness to civilization hinges on talent. 2. Urban-rural transformation is important, and talent is the key. 3. The rejuvenation of the countryside depends on human resources.= 4. The development of “back-to-the-country” is dependent on the talent. 5. Rural revitalization requires an integrated approach in terms of human resources. 6. The country’s rural振兴, human resources are the key. 7. The rural revitalization program, which has become an important part of China’s "Three Major Programs", needs an innovative mind to turn the tide of development. 8. The rural revitalization program aims to solve problems such as inadequate infrastructure, and there is a need for more professional and skilled people.

Table 4.5 Examples of translation hypotheses from the test set of WMT’21 Chinese-English. Translations in blue are the final outputs from MBR decoding.

Decoding	$\|H\|$	BLEU	BLEURT
Beam search	1	19	62.19
MBR	8	12.91	61.27
	16	13.65	62.86
	32	14.16	63.98
	64	14.01	64.85
MBR (oracle)	8	14.55	63.51
	16	15.5	65.38
	32	16.72	66.94
	64	17.32	68.16

Table 4.6 Performance of beam search, MBR decoding with temperature sampling, and MBR with temperature sampling in the oracle mode. Results are evaluated using BLOOM (7.1B) on WMT21 zh-en after post-processing.

Based on the observations, we hypothesize that the performance of MBR decoding with temperature sampling can significantly benefit from scaling up the hypothesis set, while MBR with beam sampling is not likely to achieve significant performance gain from scaling.

We validate the first part of the hypothesis by conducting further evaluations using the BLOOM model on WMT21 Chinese-English, where we vary the size of the hypothesis set from a range of 8 to 64. In addition, in order to assess the upper bound of performance with MBR using temperature sampling, we include the results from the oracle mode. In the oracle mode, we select the translation from the hypothesis set that has the highest BLEURT score with the reference translation. It is worth noting that the results from the oracle mode only serve for reference, as in a real-world setting we do not have access to the reference translations.

The results are shown in Table 4.6. We can see that MBR begins to outperform beam search starting from 16 hypotheses, in terms of the BLEURT score. As we increase $|H|$, the performance of both MBR and MBR in oracle mode continues to improve. The improvement of results from the oracle mode indicates that the BLOOM model is able to generate better translations with temperature sampling given more trials. The improvement of MBR results then indicates that MBR decoding is able to select better translation given a more comprehensive hypothesis set, possibly by calculating more accurate risks. We will explore the behavior of MBR decoding as H gets larger in the next section.

Yet, there are two key issues that can be seen from Figure 4.2. The first one is that the performance trajectories of both MBR decoding and MBR in the oracle mode, as a function of $|H|$, resemble logarithmic growth. This suggests the possibility of reaching a saturation

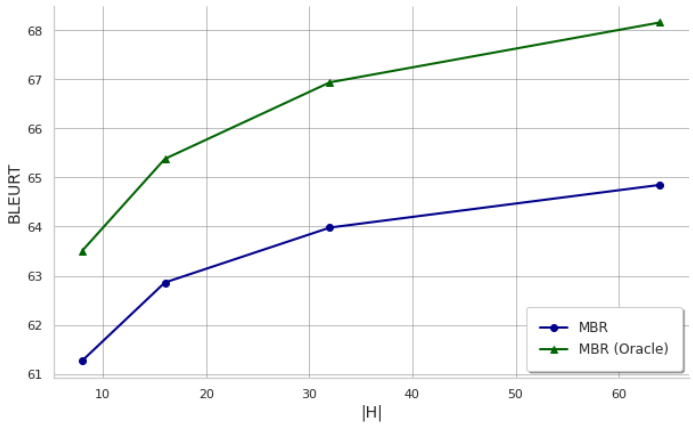


Fig. 4.2 BLEURT score on the test set of WMT'21 Chinese-English as a function of $|H|$. This figure contains the same results for MBR and MBR (oracle) as in Table 4.6.

point in performance improvement simply by scaling up the hypothesis set. The second issue is that there is a discernible gap between the BLEURT scores from MBR and MBR in the oracle mode, suggesting potential space for improvement of MBR decoding. However, bridging this gap is not trivial and we leave further improvement on MBR decoding method to future works.

4.1.3 The Behavior of MBR Decoding

Quantitative Measurements

Now we have empirically shown that MBR decoding achieves better translation performance than beam search. However, the question of how MBR generates better translations remains unexplored. We approach this question by measuring how accurate MBR decoding ranks all the hypotheses based on their risks, with respect to the BLEURT score between each hypothesis and the reference translation. The purpose of this measurement is to understand the mechanism behind MBR decoding, and gain insight into the preference pair construction for our preference optimization algorithm discussed later.

Specifically, we propose to measure the percentage of reverse pairs. We define reverse pair in the following way. Given a pair of hypotheses $(\mathbf{y}_1, \mathbf{y}_2)$ for a query sentence where $\mathbf{y}_1$ has lower Bayesian risk than $\mathbf{y}_2$, they form a reverse pair if $\text{BLEURT}(\mathbf{y}_1, \mathbf{y}_{\text{ref}}) < \text{BLEURT}(\mathbf{y}_2, \mathbf{y}_{\text{ref}})$ where $\mathbf{y}_{\text{ref}}$ is the reference translation for the query. Following the definition from Section 3.2.1, P is the preference dataset consisting of triplets $(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$. We find the set of reverse pairs R based on P as

$$R(P) = \{(\mathbf{y}_1, \mathbf{y}_2) | (\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2) \sim P, \text{BLEURT}(\mathbf{y}_1, \mathbf{y}_{\text{ref}}) < \text{BLEURT}(\mathbf{y}_2, \mathbf{y}_{\text{ref}})\} \tag{4.1}$$

$\|H\|$	$r(P_{\text{single}})$ (%)↓	$r(P_{\text{all}})$ (%)↓
8	3.54	22.8
16	0.82	21.42
32	0.21	20.74
64	0.26	20.52

Table 4.7 Percentage of reverse pairs calculated with two types of preference pairs based on the test set of WMT'21 Chinese-English.

and our proposed measure, percentage of reverse pairs, is defined as

$$r(P) = \frac{|R(P)|}{|P|} \tag{4.2}$$

Essentially, a reverse pair indicates that MBR decoding made a mistake in ranking the two hypotheses, and low $r(P)$ suggests that MBR decoding is performing well, agreeing with the quality of hypothesis translations based on BLEURT. We thus measure MBR's behavior using BLOOM on WMT21 Chinese-English, following the previous setting of using temperature sampling. We construct two preference datasets from the MBR outputs. P_{single} consists of pairs of hypotheses with lowest and highest risks. And P_{all} consists of all possible preference pairs. Table 4.7 shows that, as we increase the size of H, the percentage of reverse pairs decreases, aligning with the BLEURT scores reported in Table 4.6. In the single-pair case, MBR decoding achieves extremely high accuracy in distinguishing the quality of the two hypotheses. When considering all possible pairs, MBR is making much more mistakes. However, we can still see the benefits of increasing $|H|$ and MBR makes reasonable rankings over all the hypotheses.

Qualitative Error Analysis

Although MBR decoding gives impressive performance compared to beam search, especially when we have a large hypothesis set, there are still some issues that can be seen from the translation examples in Table 4.5. Specifically, there are three main problems: hallucination, missing information, and off-target translation. We define these three issues in a qualitative way and Table 4.8 demonstrates examples of each issue.

1. **Hallucination** is when the model generates information in the translation that is not present in the source sentence. Although the source of hallucination is hard to track, a likely reason is that the additional information is incorrectly drawn by the model from the pre-training data.

Hallucination & missing information	**Source**: 同时，省总与人社部门和智联招聘等各类互联网招聘平台对接，将招聘信息推送给困难职工，并提供就业咨询、就业指导等服务。 **Reference**: At the same time, the Yunnan Province Central Party Committee, the Ministry of Human Resources and Social Security, zhaopin.com, and other online job recruitment sites worked together to push job recruitment information to the employment-challenged, provide employment consulting and guidance, and provide other services. **MBR**: At the same time, the Provincial Federation of Trade Unions and the provincial Department of Personnel and the major Internet recruiting platforms have been paired up to send vacancy information to the workers with difficulties, and provide counseling, guidance and other services.
	Source: 当地时间7月26日，爱沙尼亚国家广播电视台报道，在德国柏林动物园附近，一辆爱沙尼亚牌照的奔驰车在26日清晨驶离车道，造成七人受伤。 **Reference**: On July 26 (CET), Estonian Public Broadcasting (ERR) reported that near the Berlin Zoo in Germany, a Mercedes-Benz with an Estonian license drove out of the driveway in the early morning of the 26th, injuring seven people. **MBR**: On July 26, the Estonian national TV reported that a Mercedes-Benz car driven by a driver from Estonia drove off the road, causing seven people to be injured in Berlin, Germany.
Off-target translation	**Source**: 新华社成都7月26日电题：把优秀返乡农民工打造成乡村振兴生力军 **Reference**: Xinhua News Agency, reported July 26th Headline: Outstanding returning rural migrant workers can be a force for rural revitalization - Xinhuanet **MBR**: 新华社记者赵超、高敬、周相吉

Table 4.8 Examples of MBR decoding outputs using BLOOM (7.1B) on WMT21 zh-en, with 64 hypotheses generated using temperature sampling. For the first two cases, texts written in blue indicate missing information and texts written in red indicate hallucination. For the last example, red texts indicate off-target translation.

2. **Missing information** is the opposite of hallucination. It happens when the model misses information present in the source sentence. The missing information ranges from a word to a sentence fragment.

3. **Off-target translation** is when the model generates translation not in the desired target language. This issue ranges from one word in the wrong language to the entire translation in the wrong language. The most common case is when the model copies words directly from the source sentence, like the 6th hypothesis from the second example in Table 4.5.

Solving the first two issues is not trivial, as it might need better model architecture and pre-training procedure. The off-target translation issue can be potentially solved by incorporating a language detection model into the MBR decoding process. We leave to future works to solve these three issues as it is out of the scope of this project.

4.2 Unsupervised Preference Optimization with MBR

In the previous section, we investigated the effectiveness of MBR decoding, showing that it achieves impressive translation performance under the in-context learning with MLLMs framework. However, as discussed previously, MBR decoding is slow and complex, so it might not be a desirable decoding method especially in a real-world setting. This motivates us to utilize preference optimization that teaches the model to imitate the MBR behavior.

Throughout the investigation of this method, there are two main research questions we aim to explore:

1. Can we teach the MLLM to learn the MBR preference, as indicated by the Bayesian risk, over translations?

2. Does preference learning with MBR generalize to translation performance?

In the following sections, we first verify that MBR decoding improves the performance of a Supervised Fine-tuning (SFT) MLLM. Next, we verify that our unsupervised fine-tuning algorithm teaches the model to learn the preference by examining training statistics. Then we explore the effects of different construction methods for the preference dataset, in terms of the translation performance of the model after preference learning. Finally, we investigate the efficacy of simultaneous preference optimization and language modelling.

Decoding	$\|H\|$	WMT'21 zh-en		WMT'21 de-en	
		BLEU↑	BLEURT↑	BLEU↑	BLEURT↑
Beam search	—	16.39	59.55	18.7	59.43
MBR	8	14.33	61.92	16.65	60.77
	16	14.36	62.17	17.18	61.89
	32	14.96	63.47	17.64	62.77

Table 4.9 BLEU and BLEURT scores on the WMT'21 Chinese-English (zh-en) and German-English (de-en) test sets. The model used is BLOOMZ (7.1B) without preference optimization. The beam width for beam search is 4, and the number of hypotheses for MBR decoding is labelled as $|H|$.

4.2.1 MBR Decoding with BLOOMZ

For this part of the project, we switch to the BLOOMZ (7.1B) model [21], which is an SFT model based on the BLOOM (7.1B) model. We denote this BLOOMZ model as π_{SFT} To verify that BLOOMZ benefits from MBR decoding, we follow the same in-context learning methodology used in Section 4.1.2 and evaluate BLOOMZ on WMT'21 Chinese-English and German-English with beam search and MBR decoding. The 8 demonstration examples are randomly selected from the validation set and are fixed for all queries. These results are reported in Table 4.9. We can see the same trend as before: MBR decoding consistently outperforms beam search and the performance keeps improving as we enlarge the hypothesis set. We use the results obtained from beam search as the baseline for the following fine-tuning experiments, and set the results obtained from MBR decoding as our goal. Our objective is that, after fine-tuning, the performance of the new model with beam search, is able to beat the performance of π_{SFT} with beam search, and achieve roughly the same performance of π_{SFT} with MBR decoding.

4.2.2 Training Statistics of Preference Optimization

We start with the simplest preference dataset that only contains one preference pair for each source sentence $\mathbf{x}$ in the validation set of WMT'21 Chinese-English. The preference pair $(\mathbf{y}_w, \mathbf{y}_l)$ contains the hypotheses with the lowest and the highest risks respectively. We initialized our language model policy π_θ and reference policy π_{ref} from the SFT model π_{SFT}, which is BLOOMZ, and train π_θ on the single-pair preference dataset P_{single} using the DPO

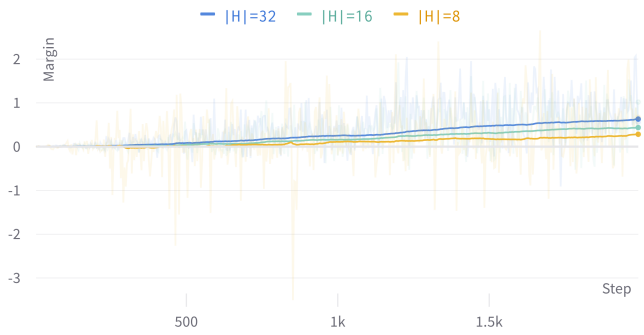


Fig. 4.3 Reward margin as a function of the number of training steps, which is the number of preference pairs the model has been trained on.

loss discussed in Section 2.3.2 for one epoch.

$$L_{\text{DPO}} = -\mathbb{E}_{(\mathbf{x},\mathbf{y}_w,\mathbf{y}_l)\sim P_{\text{single}}}[\log\sigma(\beta\log\frac{\pi_\theta(\mathbf{y}_w|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_w|\mathbf{x})} - \beta\log\frac{\pi_\theta(\mathbf{y}_l|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_l|\mathbf{x})})]$$

In order to verify whether preference optimization successfully teaches the model to learn MBR preference, we visualize the reward margins during the training process, which is defined as:

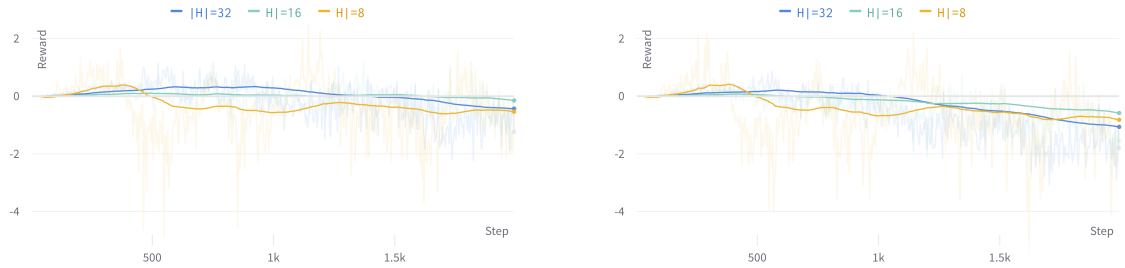
$$M = \beta\log\frac{\pi_\theta(\mathbf{y}_w|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_w|\mathbf{x})} - \beta\log\frac{\pi_\theta(\mathbf{y}_l|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_l|\mathbf{x})} \tag{4.3}$$

Positive reward margin indicates that the model π_θ is putting more probability mass on the winning hypothesis, which then suggests successful preference learning. The larger the margin, the better π_θ learns to prefer the winning hypothesis over the losing one. The visualization of reward margins is shown in Figure 4.3 for models trained on preference datasets constructed with 8, 16, and 32 hypotheses. We can see that the reward margins, after smoothing, consistently increase and stay positive as the training proceeds, for all three models.

To understand how the reward margins increase, we then visualize the rewards of the winning hypotheses and the rewards of the losing hypotheses throughout the training process, which are the two terms in the reward margin.

$$\textit{Reward}_w = \beta\log\frac{\pi_\theta(\mathbf{y}_w|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_w|\mathbf{x})},\ \textit{Reward}_l = \beta\log\frac{\pi_\theta(\mathbf{y}_l|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_l|\mathbf{x})} \tag{4.4}$$

We plot these two types of reward and show them in Figure 4.4. We can see that the rewards of the losing hypotheses decrease for all three models. However, the rewards of the winning hypotheses also show decreasing trend during training. This suggests that the model being



(a) $Reward_w$ (b) $Reward_l$

Fig. 4.4 Rewards of the winning hypotheses (a) and rewards of the losing hypotheses (b), plotted as a function of the number of training steps.

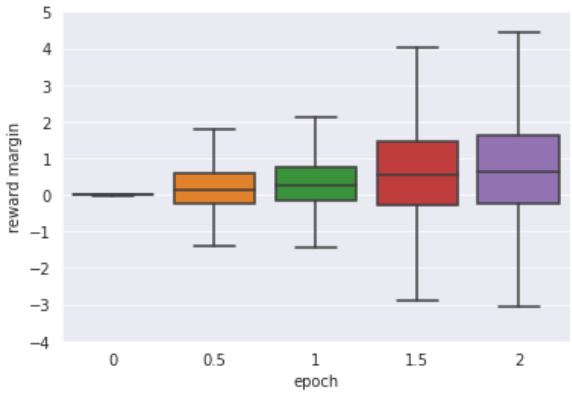


Fig. 4.5 Distributions of reward margin over all preference pairs in the held-out single-pair preference dataset. The distributions are evaluated with model checkpoints trained every 0.5 epoch. Outliers are not plotted for brevity.

trained, π_θ, learns to assign lower probability values to the winning hypotheses, which might be a problem for it to generate better translations. We give an attempt to address this issue later in Section 4.2.4.

For a more robust investigation into the success of preference learning, we also visualize the distribution of reward margin on a held-out set, which is the validation set of WMT'20 Chinese-English. We use the same single-pair selection method to construct the held-out preference dataset, and visualize the reward margins for the model trained on P_{single} with $|H| = 8$. Here we extend the training process for 2 epochs and the reward margins over all pairs in the held-out set are recorded every 0.5 epoch, shown in Figure 4.5. we can see that the median of the distributions also increases consistently as the training proceeds, which indicates that the preference learning generalizes to the held-out set. However, one issue is that the variance of the reward margin distributions also increases, especially during the second training epoch.

$\|H\|$	BLEURT($\mathbf{y}_w, \mathbf{y}_{\text{ref}}$)↑	$r(P_{\text{single}})$ (%)↓
8	64.32	7.38
16	65.15	3.26
32	65.79	1.41

Table 4.10 Statistics of the single-pair training preference datasets constructed based on the validation set of WMT'21 Chinese-English.

To summarize, we have shown evidence that unsupervised preference optimization with MBR is able to train the SFT model to prefer translation hypothesis with lower risk assigned by MBR decoding. This provides a positive answer to our first research question. In the next two sections, we explore different pair selection methods and alternative training objective, to find answer to our second research question. The training statistics for the following experiments, unless specified, show very similar pattern as the ones presented in this section, so we will not include them for brevity.

4.2.3 The Efficacy of Pair Selection Methods

In this section, we aim to explore the question of whether preference optimization generalizes to the translation performance of the model.

The Efficacy of Single Preference Pair

In Section 4.1.3, we showed that the preference of MBR decoding aligns with the translation quality assessed by BLEURT on P_{single}, as the percentage of reverse pairs $r(P_{\text{single}})$ decreases when $|H|$ gets larger. Therefore, we hypothesize that this percentage of reverse pairs is an indicator of translation performance. The model that is trained on preference dataset constructed with larger $|H|$ is likely to generate better translation during test time, and vice versa.

First, we measure the percentage of reverse pairs on our training preference dataset, specifically for the Chinese-English language pair. The results are presented in Table 4.10. We can see that the percentage of reverse pairs, under the single-pair setting, decreases as H gets larger, aligning with the BLEURT scores of the winning hypotheses.

Then we evaluate our three models that were trained on the single-pair preference datasets from the previous section, as well as three models trained on German-English preference datasets in the same manner. Table 4.11 shows the results of the trained models on the respective test sets with beam search as the decoding method. There are two things we can see. First, the training is effective as all the models outperform the baseline BLEURT

Dataset	$\|H\|$	BLEU↑	BLEURT↑
WMT'21 zh-en	8	15.59	60.5
	16	**15.73**	**60.65**
	32	15.4	59.56
WMT'21 de-en	8	18.43	60.46
	16	**18.79**	**60.68**
	32	18.58	60.44

Table 4.11 Results on the test sets of WMT'21 Chinese-English and WMT'21 German-English respectively. Models are trained on single-pair preference datasets constructed with different sizes of hypothesis set, based on the respective validation sets from WMT.

scores (59.55 for Chinese-English and 59.43 for German-English). Second, the results do not support our hypothesis that smaller $r(P_{\text{single}})$ indicates better translation performance. Specifically, on WMT'21 zh-en, though the model trained on preference dataset with $|H| = 16$ achieves the best performance, it is not significantly better than when $|H| = 8$. And the model performance drops significantly when we increase $|H|$ to be 32 and barely beats the baseline. We continue to explore using the percentage of reverse pairs as performance indicator in the next section.

The Efficacy of Training with Consecutive Pairs

We have shown that preference optimization on single-pair preference dataset is effective to improve the translation performance of the model. However, a single pair selected from the whole hypothesis set may not provide enough information on the preference of MBR decoding. We now extend this setting by using more pairs for the model to learn the MBR preference ranking more extensively. To elaborate, we experiment on the consecutive pairs strategy with $|H| = 8$ and stride of 1 on the Chinese-English dataset and construct P_{Con}. We report the dataset statistics (percentage of reverse pairs) and translation performance in Table 4.12 (first row). We can see that, the performance is worse than the baseline and there is no improvement from the second training epoch. A possible reason is that the percentage of reverse pairs is too large, which causes the model to prefer low-quality translations, as assessed by BLEURT, over high-quality ones.

Motivated by this reason, we increase the size of $|H|$ as well as the stride. For a fair comparison, we scale both $|H|$ and stride by the same factor to control for the total number of preference pairs being trained on. The results are shown in the last three rows of Table 4.12.

There are three observations we can make. Firstly, training for an additional epoch improves the model's performance. As we can see from Figure 4.6 the reward margins

$\|H\|$	Stride	$r(P_{\text{Con}})$ (%)↓	1 epoch		2 epochs	
			BLEU↑	BLEURT↑	BLEU↑	BLEURT↑
8	1	38.65	13.06	57.55	11.93	56.63
8	2	31.73	12.23	57.55	**14.9**	**60.22**
16	4	29.47	**14.54**	**59.6**	14.25	60.09
32	8	28.9	12.88	58.55	13.77	59.96

Table 4.12 Results of using consecutive pairs to construct the preference dataset P_{Con} with various values of $|H|$ and stride, on the test set of WMT21 zh-en. The percentage of reverse pairs is calculated over all the consecutive pairs in the dataset. BLEU and BLEURT scores are obtained from model checkpoints after 1 and 2 epochs of training. The last three datasets contain the same amount of preference pairs.

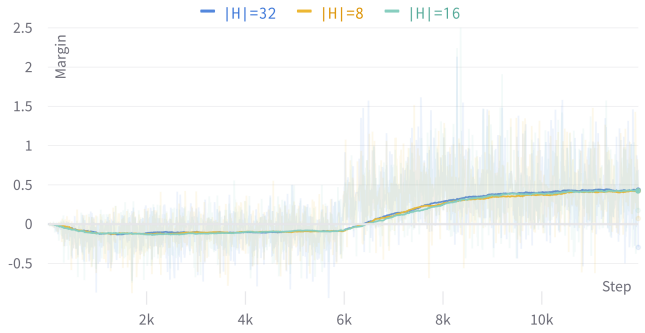


Fig. 4.6 Reward margin as a function of the number of training steps. All three models are trained for two epochs on preference datasets constructed with consecutive pairs and based on the validaton set of WMT'21 Chinese-English.

during training remain negative until the beginning of the second epoch, unlike the previous experiments. This increase in reward margins as the training proceeds suggests that the model is effectively learning the preference, and the success of learning it during the second epoch is beneficial to the translation performance as evidenced by the significant increase in BLEURT score.

Secondly, comparing the two sets of results when $|H| = 8$, it is clear that larger stride improves the performance significantly, though the preference dataset for training is smaller. This is probably due to the decrease in the percentage of reverse pairs, thus reducing the noise in the preference dataset.

However, the third observation is that, when we introduce stride, the translation performance does not have a clear relation with the percentage of reverse pairs anymore. Although the performance difference is small, this might indicate that the percentage of reverse pairs

$\|H\|$	BLEU↑	BLEURT↑
9	**17.75**	**62.23**
17	14.82	59.58

Table 4.13 Results on test set of WMT'21 Chinese-English with training on preference datasets including beam search outputs.

is not a sufficient indicator for translation performance after preference optimization, or there are other confounding variables that we are not controlling and have larger impact on performance.

The Efficacy of Training with Beam Search Outputs

We now explore the third pair selection methods that extends beyond the single-pair construction. The way is to introduce the outputs from beam search into the preference dataset. The beam search outputs are also generated from the SFT model BLOOMZ and are included in the hypothesis sets for calculating the risks. The experiments are based on WMT'21 Chinese-English and the results are shown in Table 4.13.

From the results we can see that the best performance is achieved when we add the beam search output to the hypothesis set with original size of 8. The translation performance of this model is better than not only the beam search baseline but also our goal performance (BLEURT score of 61.92). We hypothesize that the success of this method is due to two reasons. First, the beam search output serves as the anchor and training with it informs the model more on the hypothesis ranking of MBR decoding. Second, comparing the beam search output with other hypotheses might help the model specifically learn the good translations. When there is an MBR hypothesis with lower risk than the beam search output, the model learns to transfer the probability mass from the beam search output to the better hypothesis. And when there is an MBR hypothesis with higher risk than the beam search output, the model reinforces its preference on the beam search output.

However, the success of this scheme does not scale to the hypothesis set with original size of 16. In that case, the performance drops to the baseline performance. This might suggest that the model does not acquire the preference of better translation from the MBR hypotheses over the beam search output.

To summarize, we explored the efficacy of different pair selection methods on improving the translation performance of the model. Out of the three methods, adding beam search outputs to the preference dataset construction achieves the best translation performance.

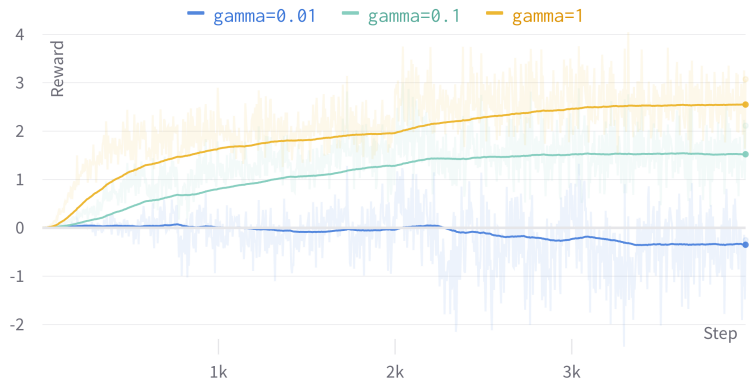


Fig. 4.7 Rewards of the winning hypotheses, plotted as a function of the number of training steps.

Regarding our second research question, the experiment results show evidence that preference learning can generalize to better translation performance, albeit under certain settings.

4.2.4 Simultaneous Preference Learning and Language Modelling

In Section 4.2.2 we noted an issue which is that the reward for the winning hypothesis showed decreasing pattern, which means that the model did not learn to increase the likelihood for the winning hypothesis during training. Now we adopt the loss function (Equation 3.6) that combines DPO loss and a language modelling loss described in Section 3.2.2.

$$L = L_{\text{DPO}}(\mathbf{x}, \mathbf{y}_w, \mathbf{y}_l) + \gamma \ln(1 + e^{w_\theta}), \text{ where } w_\theta = -\beta \log(\frac{\pi_\theta(\mathbf{y}_w|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_w|\mathbf{x})})$$

We use the single-pair preference datasets as in Section 4.2.2 to see the difference only from incorporating a language modelling objective. We visualize the rewards for the winning hypotheses for models trained with different values of γ for two epochs. Figure 4.7 shows that, with larger γ (0.1 and 1), the new training loss effectively pulls up the winning reward as it continues to increase through training.

We evaluated these trained on the test set of WMT'21 Chinese-English. Results are shown in Table 4.14 from models trained with different values of γ for one and two epochs. We can see that with proper choice of γ, adding a language modelling objective is effective, though the improvement is not significant compared to the model trained without language modelling objective (when $\gamma = 0$). Performance drop still exists after continual training for one more epoch. However, the performance drop with the mixed loss is not as severe as DPO loss only. We speculate that the language modelling objective serves as regularization to the

γ	# epochs	BLEU↑	BLEURT↑
0	1	15.59	60.5
	2	13.17	58.47
0.01	1	16.32	**61.13**
	2	15.45	59.94
0.1	1	**16.42**	60.69
	2	15.4	59.73
1	1	16.21	59.7
	2	15.53	58.97

Table 4.14 Results of using the loss function which combines preference learning and language modelling on the test set of WMT21 zh-en. γ controls the contribution of language modelling, and when $\gamma = 0$ the loss function is identical to L_{DPO}. Results are obtained from model checkpoints after 1 and 2 epochs of training.

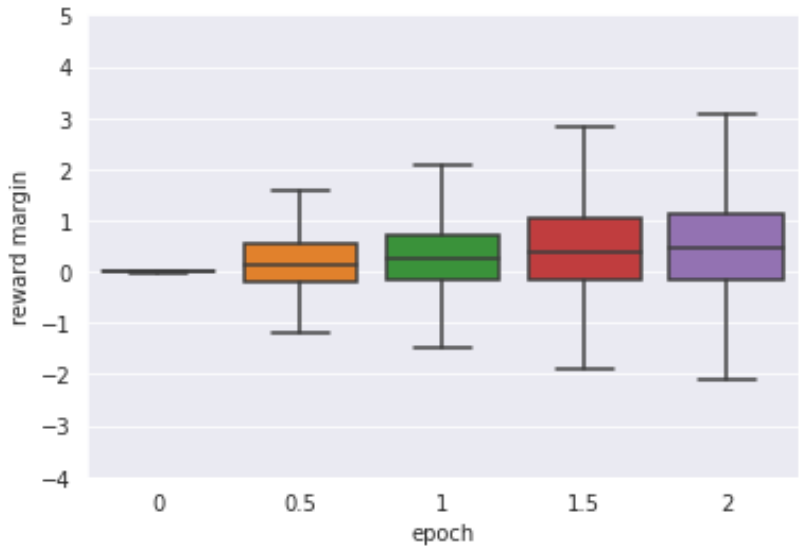


Fig. 4.8 Distribution of reward margins over the held-out validation set. The model is trained with $\gamma = 0.1$ and checkpoints are evaluated every 0.5 epoch. Outliers are not plotted for brevity.

training process. Another observation from the results is that smaller γ is more helpful. As γ decreases from 1 to 0.001, BLEURT score increases consistently.

We then look at how well the model learns the preference based on the reward margin on the held-out set. Figure 4.8 shows that while the median climbs up, the variance of reward margin is smaller than the previous experiments. This again provides evidence that the language modelling objective serves as a regularizer.

Therefore, we have shown that simultaneous preference learning and language modelling is an effective method that facilitates the generalization from preference learning to translation performance.

4.2.5 Summary

In summary, we have demonstrated that preference learning with MBR is effective to build better translation systems based on MLLMs. With our proposed fine-tuning procedure, the model is able to learn the preference of MBR decoding. And with appropriate preference pair construction, the learned preference is beneficial to the model's translation performance. However, counter-intuitively, we found that the efficacy of preference optimization does not scale with $|H|$. We leave further exploration to future works.

Chapter 5

Limitations and Future Work

Although we have demonstrated the promising aspects of our method as an efficient way to boost translation performance without the need for parallel texts, the performance improvement seen with smaller hypothesis set does not scale to large hypothesis set. Below we list a few future directions for further exploration of unsupervised preference optimization with MBR.

- **Larger monolingual dataset**: Since our method is unsupervised, it only requires monolingual texts to construct the preference dataset for training. We use the small parallel dataset from WMT mainly for diagnosis and simplicity, however, much larger monolingual datasets could be used for preference optimization with MBR.
- **More preference pairs**: Although we investigated the efficacy of including more preference pairs, there is still ways to include even more. One way is to include all possible pairs as we discussed in Section 4.2.3. Another way is to use more hypotheses to form pairs with the beam search outputs, since our method that includes beam search outputs achieved the best performance.
- **Alternative base model**: We only fine-tuned on the BLOOMZ (7.1B) model. Although the model has reasonable performance, there is a possibility that the model generated many low-quality translation hypotheses that were used to construct the preference dataset. Future work could explore other MLLMs that has been fine-tuned on supervised datasets, potentially with much more parameters, for a robust evaluation of our preference optimization algorithm.
- **Beyond Bradley-Terry model**: Our fine-tuning strategy relies on the Bradley-Terry model to formulate the preference of MBR decoding. However, this model only applies to pair-wise comparison. Since MBR decoding is able to provide a ranking list of its

preference on translations, future work could investigate using other preference models. For example, the Plackett-Luce model [25] extends the preference modelling over rankings, which makes it a suitable choice for learning the MBR preference rankings without the need for pair selection.

Chapter 6

Conclusion

In this project, we investigated in-context few-shot learning for Neural Machine Translation with Multilingual Large Language models. We have demonstrated that the number of demonstration examples is a key factor that influence the translation performance of the model, while the semantic similarity between the example and query sentence only brings trivial improvement, under the fact that the examples were drawn from a small, high-quality, and in-domain parallel dataset.

We further explored the performance of different decoding methods and showed that MBR decoding outperformed standard MAP decoding methods like greedy decoding and beam search. For MAP decoding, we showed that more exhaustive search (larger beam width) is beneficial to the performance. We further investigated the sampling methods used to generate hypotheses for MBR decoding. Through qualitative analysis, we demonstrated that temperature sampling generates more diverse translation hypotheses than beam sampling and found that the performance of the model with MBR decoding significantly improves as we increase the size of the hypothesis set.

We then investigate the behavior of MBR decoding and showed that MBR is able to rank translation hypotheses based on their quality as assessed by the BLEURT metric. However, through case studies, we found that MBR outputs suffered from hallucination, missing information, and off-target translation.

We explored the effectiveness of our proposed unsupervised preference optimization with MBR. By analyzing the reward margins over the training and held-out validation sets, we demonstrated that our method is able to teach the model to learn the ranking of translations provided by MBR decoding. We implemented and experimented with multiple pair selection methods that are used to construct the training preference dataset, and found that including beam search output into the MBR hypothesis set results in the best translation performance of the trained model. We further investigated the efficacy of simultaneous

preference optimization and language modelling and showed that this training objective is beneficial for translation performance.

Although our results showed that our proposed preference optimization method is effective at improving model's translation performance, the performance gain does not generalize to larger hypothesis sets for preference dataset construction. Future works could investigate further into different design choices of this method and increase the scale of the preference optimization process.

References

[1] Agrawal, S., Zhou, C., Lewis, M., Zettlemoyer, L., and Ghazvininejad, M. (2023). In-context examples selection for machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8857–8873, Toronto, Canada. Association for Computational Linguistics.

[2] Akhbardeh, F., Arkhangorodsky, A., Biesialska, M., Bojar, O., Chatterjee, R., Chaudhary, V., Costa-jussa, M. R., España-Bonet, C., Fan, A., Federmann, C., Freitag, M., Graham, Y., Grundkiewicz, R., Haddow, B., Harter, L., Heafield, K., Homan, C., Huck, M., Amponsah-Kaakyire, K., Kasai, J., Khashabi, D., Knight, K., Kocmi, T., Koehn, P., Lourie, N., Monz, C., Morishita, M., Nagata, M., Nagesh, A., Nakazawa, T., Negri, M., Pal, S., Tapo, A. A., Turchi, M., Vydrin, V., and Zampieri, M. (2021). Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

[3] Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[4] Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., Chen, C., Olsson, C., Olah, C., Hernandez, D., Drain, D., Ganguli, D., Li, D., Tran-Johnson, E., Perez, E., Kerr, J., Mueller, J., Ladish, J., Landau, J., Ndousse, K., Lukosuite, K., Lovitt, L., Sellitto, M., Elhage, N., Schiefer, N., Mercado, N., DasSarma, N., Lasenby, R., Larson, R., Ringer, S., Johnston, S., Kravec, S., Showk, S. E., Fort, S., Lanham, T., Telleen-Lawton, T., Conerly, T., Henighan, T., Hume, T., Bowman, S. R., Hatfield-Dodds, Z., Mann, B., Amodei, D., Joseph, N., McCandlish, S., Brown, T., and Kaplan, J. (2022). Constitutional ai: Harmlessness from ai feedback.

[5] Bradley, R. A. and Terry, M. E. (1952). Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

[6] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

[7] Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

[8] Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., Garcia, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A. M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Diaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K., Eck, D., Dean, J., Petrov, S., and Fiedel, N. (2022). Palm: Scaling language modeling with pathways.

[9] Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. (2017). Deep reinforcement learning from human preferences. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[10] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

[11] Freitag, M., Rei, R., Mathur, N., Lo, C.-k., Stewart, C., Avramidis, E., Kocmi, T., Foster, G., Lavie, A., and Martins, A. F. T. (2022). Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

[12] Garcia, X., Bansal, Y., Cherry, C., Foster, G., Krikun, M., Feng, F., Johnson, M., and Firat, O. (2023). The unreasonable effectiveness of few-shot learning for machine translation.

[13] Holtzman, A., Buys, J., Forbes, M., and Choi, Y. (2019). The curious case of neural text degeneration. *CoRR*, abs/1904.09751.

[14] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). Lora: Low-rank adaptation of large language models.

[15] Johnson, J., Douze, M., and Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

[16] Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1700–1709. ACL.

[17] Kocmi, T., Federmann, C., Grundkiewicz, R., Junczys-Dowmunt, M., Matsushita, H., and Menezes, A. (2021). To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.

[18] Kumar, S. and Byrne, W. J. (2004). Minimum bayes-risk decoding for statistical machine translation. In *North American Chapter of the Association for Computational Linguistics*.

[19] Lin, X. V., Mihaylov, T., Artetxe, M., Wang, T., Chen, S., Simig, D., Ott, M., Goyal, N., Bhosale, S., Du, J., Pasunuru, R., Shleifer, S., Koura, P. S., Chaudhary, V., O'Horo, B., Wang, J., Zettlemoyer, L., Kozareva, Z., Diab, M., Stoyanov, V., and Li, X. (2022). Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

[20] Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L., and Chen, W. (2022). What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

[21] Muennighoff, N., Wang, T., Sutawika, L., Roberts, A., Biderman, S., Le Scao, T., Bari, M. S., Shen, S., Yong, Z. X., Schoelkopf, H., Tang, X., Radev, D., Aji, A. F., Almubarak, K., Albanie, S., Alyafeai, Z., Webson, A., Raff, E., and Raffel, C. (2023). Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

[22] Neco, R. and Forcada, M. (1997). Asynchronous translations with recurrent neural nets. In *Proceedings of International Conference on Neural Networks (ICNN'97)*, volume 4, pages 2535–2540 vol.4.

[23] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., and Lowe, R. (2022). Training language models to follow instructions with human feedback. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

[24] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

[25] Plackett, R. L. (1975). The analysis of permutations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 24(2):193–202.

[26] Post, M. (2018). A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

[27] Pu, A., Chung, H. W., Parikh, A. P., Gehrmann, S., and Sellam, T. (2021). Learning compact metrics for mt. In *Proceedings of EMNLP*.

[28] Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model.

[29] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer.

[30] Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

[31] Sanh, V., Webson, A., Raffel, C., Bach, S., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Raja, A., Dey, M., Bari, M. S., Xu, C., Thakker, U., Sharma, S. S., Szczechla, E., Kim, T., Chhablani, G., Nayak, N., Datta, D., Chang, J., Jiang, M. T.-J., Wang, H., Manica, M., Shen, S., Yong, Z. X., Pandey, H., Bawden, R., Wang, T., Neeraj, T., Rozen, J., Sharma, A., Santilli, A., Fevry, T., Fries, J. A., Teehan, R., Scao, T. L., Biderman, S., Gao, L., Wolf, T., and Rush, A. M. (2022). Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

[32] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms.

[33] Schwenk, H. (2008). Investigations on large-scale lightly-supervised training for statistical machine translation. In *Proceedings of the 5th International Workshop on Spoken Language Translation: Papers*, pages 182–189, Waikiki, Hawaii.

[34] Sellam, T., Das, D., and Parikh, A. P. (2020). Bleurt: Learning robust metrics for text generation. In *Proceedings of ACL*.

[35] Sennrich, R., Haddow, B., and Birch, A. (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.

[36] Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B. (2020). Megatron-lm: Training multi-billion parameter language models using model parallelism.

[37] Song, F., Yu, B., Li, M., Yu, H., Huang, F., Li, Y., and Wang, H. (2023). Preference ranking optimization for human alignment.

[38] Stahlberg, F. (2020). Neural machine translation: A review. *The Journal of Artificial Intelligence Research*, 69:343–418.

[39] Stahlberg, F. and Byrne, B. (2019). On NMT search errors and model errors: Cat got your tongue? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3356–3362, Hong Kong, China. Association for Computational Linguistics.

[40] Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

[41] Suzgun, M., Melas-Kyriazi, L., and Jurafsky, D. (2023). Follow the wisdom of the crowd: Effective text generation via minimum Bayes risk decoding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4265–4293, Toronto, Canada. Association for Computational Linguistics.

[42] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. (2023a). Llama: Open and efficient foundation language models.

[43] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. (2023b). Llama 2: Open foundation and fine-tuned chat models.

[44] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

[45] Vilar, D., Freitag, M., Cherry, C., Luo, J., Ratnakar, V., and Foster, G. (2023). Prompting PaLM for translation: Assessing strategies and performance. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15406–15427, Toronto, Canada. Association for Computational Linguistics.

[46] Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. (2022). Finetuned language models are zero-shot learners.

[47] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

[48] Wu, Z., Wang, Y., Ye, J., Wu, Z., Feng, J., Xu, J., and Qiao, Y. (2023). OpenICL: An open-source framework for in-context learning. In *Proceedings of the 61st Annual Meeting*

of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 489–498, Toronto, Canada. Association for Computational Linguistics.

[49] Xie, S. M., Raghunathan, A., Liang, P., and Ma, T. (2021). An explanation of in-context learning as implicit bayesian inference. *CoRR*, abs/2111.02080.

[50] Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (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.

[51] Yuan, Z., Yuan, H., Tan, C., Wang, W., Huang, S., and Huang, F. (2023). Rrhf: Rank responses to align language models with human feedback without tears.

[52] Zeng, J., Meng, F., Yin, Y., and Zhou, J. (2023). Tim: Teaching large language models to translate with comparison.

[53] Zhang, B., Haddow, B., and Birch, A. (2023). Prompting large language model for machine translation: A case study.

[54] Zhu, W., Liu, H., Dong, Q., Xu, J., Huang, S., Kong, L., Chen, J., and Li, L. (2023). Multilingual machine translation with large language models: Empirical results and analysis.

[55] Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. (2020). Fine-tuning language models from human preferences.

Appendix A

Additional Prompt Formats

Here we show other prompt formats that we tested. The performance with these formats are either the same as or worse than the simple prompt format we used to present results. $\{source\}$ represents the name of the source language and $\{target\}$ represents the name of the target language.

Language pair	Prompt format
Both	$\mathbf{x}^{(1)}$: $\mathbf{y}^{(1)}$ **x**:
Both	$\{source\}$: $\mathbf{x}^{(1)}$ $\{target\}$: $\mathbf{y}^{(1)}$ $\{source\}$: **x** $\{target\}$:
Both	Translate $\{source\}$: $\mathbf{x}^{(1)}$ into $\{target\}$: $\mathbf{y}^{(1)}$ Translate $\{source\}$: **x** into $\{target\}$:
de-en	Deutsch: $\mathbf{x}^{(1)}$ English: $\mathbf{y}^{(1)}$ Deutsch: **x** English:
zh-en	中文: $\mathbf{x}^{(1)}$ English: $\mathbf{y}^{(1)}$ 中文: **x** English:

Table A.1 Additional prompt format. "Both" means for both German-English and Chinese-English. We only show the case for one demonstration example for brevity.

Appendix B

Fixed Demonstration Examples

In Table B.1 and Table B.2 we show the fixed demonstration examples from the validation set of WMT'21 Chinese-English.

Source	招待会现场举行了新中国成立70周年主题图片展，播放了《中国2019》、《美丽中国》等宣传视频。
Ref	A picture exhibition with the theme of the 70th anniversary of the new China was held at the reception venue, at which some promotion videos, like China 2019 and Beautiful China , were shown.
Source	占美国经济总量约70%的个人消费支出当季环比增幅为4.6％，拉动经济增长3.03个百分点。
Ref	Personal consumption expenditures, which accounted for about 70% of the U.S. economy, increased by 4.6% quarter-on-quarter, driving 3.03 percentage points of economic growth.
Source	中国和广大非洲国家都是发展中国家，我们要坚持把经济社会发展放在第一位，相互支持各自选择的发展道路。
Ref	China and African countries are both developing countries, we must persist in putting economic and social development first and mutually support the development path chosen by each nation.
Source	经济运行更稳定、增长质量更高、增长前景可期的中国，对世界经济发展是长期利好。
Ref	With more stable economic operations, higher growth quality, and promising growth prospects, China is beneficial to the global economic development in the long run.

Table B.1 The first four demonstration examples that are fixed for preference optimization.

Source	地下二层的和平局，青瓦石砖堆砌起的围墙和富有年代气息的照相馆、大戏院、食品店、旧书局等老店铺，处处透着浓浓的北京味儿。
Ref	Inside HePingGuoJu on the B2 floor, walls built up with grey-green roof tiles and stone bricks, as well as old-fashioned photo studios, cinemas, grocery stores, and bookstores together formed a rich vignette of old Beijing.
Source	美国参议院批准斯卡利亚出任劳工部长-新华网
Ref	US Senate Approves Scalia as Labor Secretary - Xinhuanet
Source	但是，二季度私人部门投资较上季度下滑6.3％，拖累当季经济增长1.16个百分点；出口较上季度下滑5.7％，净出口拖累当季经济增长0.68个百分点。
Ref	However, private sector investment in the second quarter fell by 6.3% from the previous quarter, dragging down economic growth by 1.16 percent; exports fell by 5.7% from the previous quarter, and net exports dragged down economic growth by 0.68 percent.
Source	庆祝中华人民共和国成立70周年外国专家招待会举行 韩正出席并致辞
Ref	Foreign Expert Reception to Celebrate the 70th Anniversary of the Founding of the People's Republic of China was held, Han Zheng attended and addressed the Reception.

Table B.2 The last four demonstration examples that are fixed for preference optimization.