Using Reinforcement Learning LoRA For Interactive Machine Translation Systems

Anonymous ACL submission

Abstract

The use of large language models (LLMs) is growing due to their impressive performance on a wide range of tasks. As new versions of these models appear to achieve better results, their size often increases, making it more challenging to maintain different versions specialized in specific domains. However, by employing the Low-Rank Adaptation (LoRA) method, we can bypass this space limitation, as the fine-tuning changes of the model are stored in a file of just a few megabytes. In the Machine Translation (MT) field, it is common to have models specialized for particular domains or language pairs. In our case, we apply these models within Interactive Machine Translation (IMT), where it is crucial that the model generates high-quality translations and adapts to user modifications. We have incorporated Reinforcement Learning (RL) techniques to optimize the model using various metrics to enhance this adaptability further. Our results demonstrate that these methods effectively improve the quality of translations generated by the models, though in some cases, this comes at the cost of a slight reduction in generalization capability.

1 Introduction

Machine Translation (MT) has undergone significant changes in recent years, mainly due to the advent of neural models. These advances have enabled models to perform with a level of efficiency comparable to that of human translators across a broad range of machine translation tasks (Toral, 2020). Despite this progress, there are still many instances where models struggle to produce high-quality translations. Moreover, in fields where even minor errors can have serious consequences—such as medicine, law, or the military—perfect accuracy in translation is imperative. In such cases, human involvement is required for post-editing to ensure flawless translations, as experts review and correct these. Various Computer-Assisted Translation (CAT) tools have been developed to minimize the effort these human experts require, including Interactive Machine Translation (IMT) (Federico et al., 2014; Sanchis-Trilles et al., 2014; Herbig et al., 2020).

IMT systems aim to reduce the effort required by users by creating a collaborative framework where the expert user and the translation model work iteratively to produce perfect translations. Instead of correcting all the errors found, the user only needs to correct the first error and provide this feedback to the system, which then generates an improved translation. This process is repeated until the user approves the translation. Various protocols can be implemented to facilitate this interaction (Foster et al., 1997; Alabau et al., 2010; Domingo et al., 2017), but in our case, we will use the prefix-based protocol, as it aligns more closely with the generation process of MT models.

One technique employed alongside IMT systems involves providing each user with a personalized translation model, slightly adjusted to favor the user's preferred word choices. This model adjustment can be achieved through online or active learning techniques (Peris and Casacuberta, 2018, 2019), allowing the model to adapt as the system is being used in real-time. However, this approach is becoming increasingly obsolete with the advent of Large Language Model (LLM) such as GPT (Achiam et al., 2023), BLOOM (Scao et al., 2022), Gemini (Team et al., 2023), and Llama2 (Touvron et al., 2023), which are growing in size, making it impractical for each user to maintain a personal copy.

By applying the Low-Rank Adaptation (LoRA) technique (Hu et al., 2021) to LLM trained for multiple tasks, we can fine-tune the model for a specific domain without creating a new copy of the model for each case. This technique allows us to save the changes required for the model to function

in the targeted domain in only a few megabytes of file. Thus, instead of maintaining separate copies of the model for each task, we only need to keep the original model and the lightweight LoRA files, which are then added to the base model to use them. Given the minimal storage requirements and that LoRA has demonstrated comparable results to conventional fine-tuning, the possibility of each user having their customized model or maintaining multiple models for specific tasks becomes feasible once again.

In this article, we aim to evaluate the efficiency of LoRA fine-tuned models within the IMT domain. It has been demonstrated that fine-tuning large language models for specific translation tasks improves the quality of the generated translations. However, in the field of IMT, we require more than just high-quality initial translations; we need models that can adapt to user feedback, effectively generalizing to produce alternative translations that better align with the translator's expectations. Additionally, we explore how different training methods for LoRA models impact human effort metrics like WSR, KSR, or MAR. To this end, we have also implemented a Reinforcement Learning (RL) algorithm to fine-tune the models, optimizing metrics such as Accuracy, TER, and BLEU.

2 Related Work

In this article, we focus on three primary areas of research. First, we explore the use of LoRA and adapters for fine-tuning. Second, we investigate the application of these models within IMT. Finally, we examine the use of RL techniques for training these translation models.

Large Language Models A significant number of Large Language Models (LLMs) have emerged recently. Among them, we have chosen to use BLOOM (Scao et al., 2022) primarily because it is an open-source model, trained across multiple languages, and available in various sizes. While the list of prominent LLMs is constantly evolving, some of the most well-known currently include GPT-4 (Achiam et al., 2023), LLaMA2 (Touvron et al., 2023), Gemini (Team et al., 2023), FALCON (Almazrouei et al., 2023), and Mistral (Jiang et al., 2023).

Finetuning with Adapters While we are employing LoRA, there are other methods that fall under the umbrella of Parameter-Efficient Fine-Tuning (PEFT). Several of these methods also utilize adapters for fine-tuning the model, such as Low-Rank Hadamard Product (LoHA) (Hyeon-Woo et al., 2021) and Orthogonal Fine-Tuning (OFT) (Qiu et al., 2023). Other PEFT methods, categorized as Soft Prompts, aim to identify the optimal input tensor for a given task rather than altering the model's weights. Among these are techniques like prompt tuning (Lester et al., 2021), prefix tuning (Li and Liang, 2021), and P-tuning (Liu et al., 2023).

Interactive Machine Translation In the field of IMT, various protocols can be followed depending on how the user performs the corrections. In our case, we are working at the prefix level (Foster et al., 1997), requiring the user to make corrections from left to right. Alternatively, segment-level protocols (Domingo et al., 2017) allow users more flexibility as they can correct wherever words they find, though it supposes a more significant challenge for the translation model. Other methods to reduce human effort include using confidence measures (Specia et al., 2013), touch-only interactions (Wang et al., 2020), or auto-completing written predictions (Barrachina et al., 2009). These tools are often integrated into workbenchs like CasMaCat (Alabau et al., 2013) or TranSmart (Huang et al., 2021) to minimize human effort as much as possible.

Reinforcement Learning There are various approaches to incorporating RL into the training of translation models. However, most approaches begin with a pre-trained model due to the typically large action space involved. We are using the Policy Gradient (PG) algorithm (Sutton et al., 2000) to improve the model's performance on Accuracy, BLEU, and TER metrics. Additionally, other research efforts focus on aligning the evaluation metric with the training objective (Bahdanau et al., 2016), leveraging bandit feedback in reinforcement learning (Kreutzer et al., 2018), or simplifying the input provided during the IMT session at the cost of requiring an RL model that adapts and learns from the input (Lam et al., 2019).

3 System Framework

In this article, we explore two distinct areas of research. The first focuses on training LLM using the LoRA method to minimize their storage footprint. We integrated a RL algorithm, specifically PG (Sutton et al., 2000), into this training approach to optimize models for metrics pertinent to MT and

IMT, including translation accuracy, BLEU, and TER scores. Additionally, we tested these models within an IMT system to evaluate their performance and determine whether the training applied to the base model enhances its effectiveness. In the context of IMT, it is crucial not only to generate a high-quality initial translation and adapt effectively to user modifications.

3.1 Reinforcement Learning Training

Since we planned to use the models trained with LoRA in an IMT system, we wanted to evaluate the performance of models trained using the standard approach and observe how models optimized for different metrics behave within this environment. For instance, BLEU is commonly used to assess the quality of generated translations, while TER is more closely associated with the amount of post-editing required. To explore these aspects, we decided to incorporate a RL algorithm into the training process of the LoRA models.

The first step in implementing the RL algorithm is to define our objective. In our case, we aim to maximize the expected reward of following the model's policy. This can be represented as:

$$\text{maximize } \mathbb{E}_{\hat{y}_1^T \sim \pi_\theta(\hat{y}_1^T)}[r(\hat{y}_1, ..., \hat{y}_T)] \quad (1)$$

where $\pi_\theta(\cdot)$ is the policy that we are following, which is represented by our LLM, $\hat{y}_t$ is the word chosen by the model at time t and $r(\hat{y}_1, ..., \hat{y}_T)$ is the reward associated with the sequence $\hat{y}_1, ..., \hat{y}_T$.

When training using Teacher Forcing (Bengio et al., 2015), a ground truth sequence is provided, and words are selected based on the current policy. Upon generating an end-of-sequence (EOS) token, the reward is calculated by comparing the generated sequence with the ground truth. This training process aims to find the model parameters that maximize this expected reward. This loss is defined as the negative expected reward of the generated sequence:

$$\mathcal{L}_\theta = -\mathbb{E}_{\hat{y}_1^T \sim \pi_\theta(\hat{y}_1^T)}[r(\hat{y}_1), ..., \hat{y}_T] \quad (2)$$

If we use only a single sample from the action distribution from the model to approximate the expectation, the derivative of the previous function can be expressed as:

$$\bigtriangledown_\theta \mathcal{L}_\theta = -\mathbb{E}_{\hat{y}_1^T \sim \pi_\theta}[\bigtriangledown_\theta \log \pi_\theta(\hat{y}_1^T) r(\hat{y}_1^T)] \quad (3)$$

By applying the chain rule and differentiating with respect to the final softmax layer of the model, we can define this gradient as follows (Williams, 1992; Zaremba and Sutskever, 2015):

$$\frac{\partial \mathcal{L}_\theta}{\partial o_t} = \Big(\pi_\theta(y_t|\hat{y}_{t-1}, s_t, c_{t-1}) - \mathbf{1}(\hat{y}_t)\Big)\Big(r(\hat{y}_1^T) - r_b\Big) \quad (4)$$

where o_t is the input of the softmax function, $\mathbf{1}(\hat{y}_t)$ is the one-hot vector representation of the ground-truth and r_b is a baseline reward and can be any value, provided it is independent of the parameters of model.

We employed Eq. (4) to train the models using three different metrics. The first and most straightforward metric is translation accuracy. While accuracy is not typically used in the field of MT, our goal is to minimize the number of corrections required by the user in an IMT environment. Given this objective, it seemed logical to experiment with a more direct metric that provides insight into the number of correct words generated and, consequently, the number of corrections still needed.

The second metric we employed is BiLingual Evaluation Understudy (Bleu), the most commonly used metric for evaluating the quality of translations generated by MT models. Additionally, existing studies have utilized Bleu for training with RL, demonstrating a slight improvement in translation quality compared to standard training methods.

Finally, we employed the Translation Error Rate (TER) metric, which is particularly relevant in the context of IMT, as it provides insight into the number of operations —insertions, substitutions, deletions, and swaps—required to correct a translation in a post-editing environment.

3.2 IMT Implementation

The Neural Machine Translation (NMT) framework operates as follows. Given a source language sentence $x_1^J = x_1, \ldots, x_J$, the goal is to generate the most probable translation $\hat{y}_1^{\hat{I}} = \hat{y}_1, \ldots, \hat{y}_{\hat{I}}$ in the target language Y. The fundamental equation of the statistical approach to NMT is then expressed as:

$$\begin{aligned} \hat{y}_1^{\hat{T}} &= \arg\max_{T, y_1^T} \Pr(y_1^T \mid x_1^J) \approx \\ &\approx \arg\max_{T, y_1^T} \prod_{t=1}^{T} \pi_\theta(y_t \mid y_1^{t-1}, x_1^J) \end{aligned} \quad (5)$$

		Europarl		HPLT		NLLB	
		Es–En	**Fr–En**	**Eu–En**	**Sw–En**	**Ln–En**	**Yo–En**
Train	$\|S\|$	2.0M	2.0M	606K	1.7M	2.9M	1.5M
	$\|T\|$	51.6M/49.2M	60.5M/54.5M	65.7M/62.6M	140.1M/121.6M	141.5M/128.8M	111.6M/84.9M
	$\|V\|$	422.6K/309.0K	160.0K/131.2K	725.1K/456.7K	918.7K/825.4K	483.9K/748.7K	1.2M/619.3K
Val.	$\|S\|$	3003	3000	2000	2000	2000	2000
	$\|T\|$	69.5K/63.8K	73.7K/64.8K	220.3K/211.4K	167.3K/144.5K	96.5K/88.1K	159.5K/119.1K
	$\|V\|$	16.5K/14.3K	11.5K/9.7K	13.8K/11.3K	7.7K/7.4K	6.4K/5.7K	9.7K/7.8K
Test	$\|S\|$	3000	1500	2000	2000	2000	2000
	$\|T\|$	62.0K/56.1K	29.9K/27.2K	213.7K/204.2K	161.4K/139.2K	99.9K/91.9K	155K/115.1K
	$\|V\|$	15.2K/13.3K	6.3K/5.6K	13.7K/11.1K	7.3K/7.1K	6.7K/5.9K	9.9K/7.7K

Table 1: Corpora statistics. K denotes thousands and M millions. $|S|$ stands for number of sentences, $|T|$ for number of tokens and $|V|$ for size of the vocabulary. **Fr** denotes French; **Es**, Spanish; **Eu**, Basque; **Sw**, Swahili; **Ln**, Lingala; **Yo**, Yoruba; and **En**, English;

where $\Pr(y_1^T|x_1^J)$ and $\pi_\theta(y_t|y_1^{t-1}, x_1^J)$, are the probability distribution and the probability that assigns the policy to the next target word given the source sentence and the previous words so far.

We have developed a prefix-based IMT system integrated with the NMT framework. Upon receiving a translation from the system, the user provides feedback by correcting the first detected error f_p. The system then leverages this feedback to generate the subsequent translation with the highest probability, ensuring it maintains the same prefix and incorporates the user-provided correction. This iterative process continues until the user fully validates the sentence. The translation procedure can be formally described by incorporating the feedback and the last generated hypothesis into 5 as follows:

$$\hat{y}_1^{\hat{T}} \approx \arg\max_{T, y_1^T} \prod_{t=1}^{T} \pi_\theta(y_t \mid y_1^{t-1}, x_1^J, \bar{y}_1^{\bar{T}}, f_1^p)$$
$$\text{subject to} \qquad (6)$$
$$1 \leq t < p \qquad f_t = y_t = \bar{y}_t$$
$$f_p = y_p \neq \bar{y}_p$$

where $\bar{y}_1^T = \bar{y}_1, \ldots, \bar{y}_T$ is the previous hypothesis, f_1^p is the feedback provided, and p is the length of the feedback.

4 Experimental Framework

4.1 Evaluation metrics

We utilized a range of evaluation metrics to assess the quality of translations produced by our models after fine-tuning them using a specific training method and language pair. This approach allows us to compare the improvement of each model across different techniques and establish their baseline performance for experiments related to IMT.

To assess the quality of the translations, we have computed the following metrics by using the implementation from *sacreBLEU*[1] (Post, 2018):

BiLingual Evaluation Understudy (Papineni et al., 2002): computes the geometric mean of the modified n-gram precision, adjusted by a brevity penalty to account for short sentences. This adjustment ensures the consistency of BLEU scores across different translation outputs.

Translation Error Rate (Snover et al., 2006): calculates the number of word-level edit operations—insertions, substitutions, deletions, and swaps—normalized by the total word count in the final translation. This metric is a simplified approximation of the user effort required to correct a translation hypothesis in a traditional post-editing scenario.

Given that these models are intended for use within the field of IMT, it is crucial to assess the human effort required to correct the translations they produce using a prefix-based IMT environment. We have simulated this process, and its methodology is detailed in Section 4.4.

To assess the human effort performed to correct the translations, we have computed the following metrics:

Word Stroke Ratio (Tomás and Casacuberta, 2006): quantifies the number of words which

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

SOURCE: Hartu edalontzia mahaira hutsik.
TARGET: Take your glass back to the table empty .

ITER-0	Translation hypothesis	Leave the door open .
ITER-1	Feedback Translation hypothesis	**Take** *Take* your glass again to the table empty
ITER-2	Feedback Translation hypothesis	*Take your glass* **back** *Take your glass back* to the table .
ITER-3	Feedback Translation hypothesis	*Take your glass back to the table* **empty** *Take your glass back to the table empty .*
END	Final translation	*Take your glass back to the table empty .*

Figure 1: Prefix-based IMT session for translating a sentence from Basque to English, the process begins with the system providing an initial hypothesis.

must be changed, normalized by the total word count in the final translation.

Key Stroke Ratio (Tomás and Casacuberta, 2006): quantifies the number of characters wich must be changed, normalized by the number of character in the final translation.

Mouse Action Ratio (Barrachina et al., 2009): quantifies the number of mouse actions performed, normalized by the number of characters in the final translation.

Lower values in these metrics indicate better system performance, as fewer user interactions are required to achieve perfect translations. Furthermore, when comparing results, we should prioritize reducing the keyboard effort, as some systems have implemented automated mouse interactions or the use of alternative devices for system navigation, which directly reduces the number of mouse actions by other means.

4.2 Corpora

In our experiments, we utilized language pairs that are included in the extensive BLOOM language model. We selected languages with varying levels of representation within the dataset used to train this model. The languages chosen for our experiments are Spanish (*es*), French (*fr*), Basque (*eu*), Swahili (*sw*), Lingala (*ln*), and Yoruba (*yo*), with translations occurring between these languages and English (*en*). Among these, Spanish, French, and English have the highest representation in the dataset used in BLOOM, followed by Basque and Swahili. Lingala and Yoruba have the most miniature representations. The dataset used in our experiments was sourced entirely from three different origins

For Spanish and French, we used the Europarl corpus (Koehn, 2005), a compilation of proceedings from the European Parliament. We employed the High Performance Language Technologies (HPLT) corpus (De Gibert et al., 2024) for Basque and Swahili, which was extracted from the internet using web crawlers and subsequently post-processed. Lastly, we utilized the No Language Left Behing (NLLB) corpus (Costa-jussà et al., 2022) for Lingala and Yoruba, designed to include as many languages as possible while maintaining high data quality.

Table 1 shows the main features of the corpus.

4.3 Systems

We started with the open-source LLM BLOOM (Scao et al., 2022) to train our models. BLOOM is a decoder-only transformer model (Vaswani et al., 2017) that has been trained on a dataset comprising 46 spoken languages and 13 programming languages. The base LLM model consists of 176 billion parameters, which poses a challenge due to the capacity of our GPUs. Therefore, we specifically used the checkpoint available at '*https://huggingface.co/bigscience/bloom-560m*' from the Hugging Face library (Wolf et al., 2020) which consists of 560 million parameters. This checkpoint was chosen primarily due to GPU memory constraints and because it yielded high-quality translation results in our fine-tuned models.

For fine-tuning each of the models trained in this study, we employed the LoRA technique (Hu et al., 2021). This approach significantly preserves storage space: instead of maintaining a full copy of the original model with modified values for each fine-tuned model, LoRA allows us to store a lightweight file containing only a few parameters per model. These parameters are used to calculate weights added to the ones from the original model, thereby saving substantial storage space compared with the other method.

ES-EN	5-shot	LoRA	RL Acc	RL Bleu	RL TER	5-shot	LoRA	RL Acc	RL Bleu	RL TER	EN-ES
BLEU	18.10	**26.71**	26.64	26.66	25.9	16.05	**27.01**	26.81	26.50	26.26	BLEU
TER	80.56	66.41	66.39	**66.20**	67.1	82.24	**65.56**	66.00	65.83	66.41	TER
FR-EN	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**EN-FR**
BLEU	22.33	27.34	**27.71**	26.68	25.80	16.22	29.08	**30.02**	28.83	29.73	BLEU
TER	72.97	65.90	**64.41**	66.33	67.98	79.36	71.01	69.37	69.47	**68.41**	TER
EU-EN	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**EN-EU**
BLEU	02.12	**25.78**	24.91	23.71	23.82	01.75	**17.11**	16.20	15.71	16.64	BLEU
TER	269.3	**71.18**	72.66	76.27	76.27	275.8	92.70	97.95	95.96	**91.47**	TER
SW-EN	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**EN-SW**
BLEU	08.56	47.21	**48.15**	44.87	42.99	15.11	43.26	**45.04**	40.60	41.15	BLEU
TER	172.7	55.14	**54.21**	58.37	60.59	129.4	67.26	**61.39**	71.10	71.43	TER
LN-EN	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**EN-LN**
BLEU	00.75	**05.61**	04.97	05.48	04.94	00.57	02.88	**03.09**	02.90	02.00	BLEU
TER	**119.3**	147.5	151.7	136.3	146.5	**111.9**	238.8	228.8	212.7	265.8	TER
YO-EN	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**EN-YO**
BLEU	00.24	02.72	**02.95**	01.62	02.91	00.01	**00.80**	00.68	00.74	00.63	BLEU
TER	**137.6**	190.2	178.3	217.1	159.4	411.4	**390.0**	453.0	418.2	429.5	TER

Table 2: Quality results of the translations generated by our trained models compared to performing 5-shot on the base model. All values are reported as percentages. Best results are denoted in bold. **Fr** denotes French; **Es**, Spanish; **Eu**, Basque; **Sw**, Swahili; **Ln**, Lingala; **Yo**, Yoruba; and **En**, English;

For the LoRA method, we reduced the matrix dimensionality to $r = 8$ and applied the method to the transformer's query, key, and value layers from the attention blocks. The models were fine-tuned over $100,000$ steps, using a batch size of 8 and a learning rate of $2e-3$.

In total, we trained four different models for each language pair. The first model, referred to as **LoRA**, was trained using the LoRA method with the configuration outlined previously. The other three models were trained using the Reinforcement Learning algorithm described in Section 3.1. These include a model trained to maximize translation accuracy (**RL Acc**), another optimized for the BLEU metric (**RL Bleu**), and finally, a model where the goal was to maximize the TER metric (**RL TER**).

4.4 Simulation

We used simulated users to conduct experiments and evaluate the models to address the significant time and financial costs associated with human evaluation during the development phase. This choice allowed us to establish a more controlled experimental environment by minimizing potential external errors and removing the human factor. These simulated users were responsible for generating accurate translations from a given reference and providing feedback to the IMT system.

To conduct these evaluations, we employed the prefix-based protocol outlined by Foster et al. (1997), where the user identifies and corrects the leftmost incorrect word, validating all preceding words in the prefix up to the point of correction. Thus, the validated prefix includes all words preceding and including the corrected term.

We have opted for the prefix-based protocol as it aligns more effectively with the generation procedure of LLMs, which generate words from left to right. This approach allows us to incorporate all the validated words from the prefix into the prompt provided to the LLM, ensuring that the translation continues seamlessly.

At the start of the simulation, the system generates an initial translation hypothesis, which the simulated user then reviews. The user identifies the first error by comparing the hypothesis with the reference, examining both the words and their positions. Upon detecting an error, the user consults the reference to confirm the correct term and provides this correction as feedback to the system. Corrections are inputted via a keyboard stroke, and if the error is not immediately adjacent to the previous correction, a mouse action is also required. This process continues until the simulated user has accurately translated the entire sentence. A final mouse action is performed to validate the translation, signifying that the entire sentence has been correctly translated.

Figure 1 illustrates a simulation example for translating a source sentence from Basque to En-

ES-EN	5-shot	LoRA	RL Acc	RL Bleu	RL TER	5-shot	LoRA	RL Acc	RL Bleu	RL TER	EN-ES
WSR	51.38	54.19	53.86	**50.84**	67.55	**57.18**	70.92	61.60	74.08	79.73	WSR
KSR	54.18	56.57	56.11	**53.11**	69.20	**59.67**	73.62	64.74	76.35	81.45	KSR
MAR	23.12	19.66	19.97	20.95	**15.64**	20.81	13.85	16.61	12.88	**11.13**	MAR
FR-EN	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**EN-FR**
WSR	**51.89**	55.06	52.79	55.78	53.94	58.40	52.52	61.36	**51.06**	56.74	WSR
KSR	**53.20**	56.10	54.07	57.27	55.06	56.69	53.87	64.44	**53.33**	59.71	KSR
MAR	22.62	20.17	20.77	**20.00**	20.77	19.34	17.31	**14.34**	17.46	15.74	MAR
EU-EN	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**EN-EU**
WSR	**65.33**	85.60	88.55	86.68	89.04	**80.41**	88.19	88.00	89.23	88.83	WSR
KSR	**69.60**	88.00	89.36	87.70	89.84	**86.67**	90.15	89.74	90.75	90.76	KSR
MAR	22.98	86.51	**07.96**	08.74	08.05	16.19	09.04	09.23	**08.99**	09.19	MAR
SW-EN	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**EN-SW**
WSR	**60.34**	60.77	67.97	61.40	61.78	64.45	**54.92**	68.31	72.24	69.09	WSR
KSR	64.33	**61.99**	69.05	62.77	63.18	68.07	**54.44**	68.39	72.52	69.38	KSR
MAR	20.42	10.63	**09.59**	11.11	10.98	27.27	10.26	**08.60**	08.83	09.20	MAR
LN-EN	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**EN-LN**
WSR	79.70	72.08	71.88	72.79	**70.96**	96.72	86.75	87.64	86.22	**84.91**	WSR
KSR	84.37	77.61	77.58	78.45	**77.02**	98.23	89.71	90.87	89.57	**88.67**	KSR
MAR	83.64	24.36	24.54	24.12	**13.33**	**12.04**	16.15	15.97	16.97	17.99	MAR
YO-EN	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**5-shot**	**LoRA**	**RL Acc**	**RL Bleu**	**RL TER**	**EN-YO**
WSR	83.66	80.70	**76.63**	81.31	80.03	98.64	95.69	**93.97**	95.51	95.67	WSR
KSR	88.76	86.60	**83.51**	87.11	86.19	99.10	97.10	**95.96**	97.06	97.20	KSR
MAR	20.23	**20.15**	22.64	20.32	20.71	**08.49**	10.27	11.62	10.37	10.30	MAR

Table 3: Human Effort results of the translations generated by our trained models compared to performing 5-shot on the base model. All values are reported as percentages. Best results are denoted in bold. **Fr** denotes French; **Es**, Spanish; **Eu**, Basque; **Sw**, Swahili; **Ln**, Lingala; **Yo**, Yoruba; and **En**, English;

glish. The translation session begins with the system generating an initial hypothesis that requires user review and correction. In the first iteration, the user corrects the initial error, *Take*, prompting the system to generate a new hypothesis incorporating this feedback. During the next iteration, the user identifies and corrects a subsequent error, *back*, thereby validating the prefix *Take your glass*. This process repeats for one more iteration, with the user correcting the word *empty*, leading the system to produce the correct translation. The session concludes with the user validating the final translation through a mouse action.

5 Results

To evaluate the performance of our models, we first need to assess the quality of the translations they generate, as this will serve as the starting point for the IMT systems. We also tested the original BLOOM model using few-shot prompting to adapt it to the task for a more comprehensive comparison. Given that we provided 5 examples for each task, we refer to this baseline model as **5-shot**.

The results based on BLEU and TER scores are presented in Table 2. As expected, fine-tuning the model specifically for the language pairs used in the task proved more beneficial across all language pairs than providing five examples. It is worth noting that the quality improvement in translation was less significant for Lingala and Yoruba, the languages that were least represented in the dataset used to train the BLOOM LLM. This is partly because the LoRA method performs better when only minor adjustments are needed to adapt the model to the task.

The most significant improvements compared to the baseline were observed in Basque and Swahili. These languages, which have moderate representation in the LLM, also share similarities with more prominent languages included in the model. For Basque, using few-shot prompting resulted in a BLEU score of only 2 points; however, after applying LoRA, the quality increased dramatically to 25 points. Similarly, for Swahili, the BLEU score jumped from 8 to 48 points, marking a substantial improvement.

No significant difference in translation quality was observed between using the LoRA method and employing the Reinforcement Learning implementation, which aimed to optimize Accuracy, BLEU,

5-shot	**LoRA**
TARGET: Employment and grant RSS Back	TARGET: Employment and grant RSS Back
ITE 0: Enplegu eta Beken RSSak Itzuli	ITE 0: RSS Employment and Job Search
ITE 1: *Employment* and Business RSSak Itzuli	ITE 1: *Employment*
ITE 2: *Employment and grant* RSSak Itzuli	ITE 2: *Employment and*
ITE 3: *Employment and grant RSS* Itzuli	ITE 3: *Employment and grant*
ITE 4: *Employment and grant RSS Back*	ITE 4: *Employment and grant RSS*
	ITE 5: *Employment and grant RSS Back*

Figure 2: Example of the iterative correction process in an IMT system from Basque to English using the base model **5-shot** and the **LoRA** model is provided.

and TER metrics.

The results of applying these models in a prefix-based IMT system are presented in Table 3. This table displays the values for the WSR, KSR, and MAR for each model, with the best results for each language pair highlighted in bold. This evaluation not only assesses the quality of the initial hypothesis generated by the models but also examines their ability to adapt to user corrections by providing valid new translations that align with the feedback, thereby testing their generalization capability.

Although the initial experiment, which assessed the quality of translations generated by the models, demonstrated a clear improvement using the LoRA method, this significant enhancement is not as evident in the current context. In some cases, the baseline model using few-shot prompting achieves a more significant reduction in effort compared to our models. The most noticeable differences from the baseline are observed in languages with lower representation. Nonetheless, the reduction in effort is minimal, suggesting a slight advantage to using these models over traditional post-editing methods.

When comparing the IMT sessions of our models to the baseline, it becomes apparent why the improved quality of the models does not translate into a corresponding reduction in human effort. Figure 2 illustrates this for the Basque-English language pair. As shown in Table 2 the initial hypothesis from the LoRA model is significantly better than the generated from the 5-shot model. However, the discrepancy in results becomes evident in subsequent iterations. Despite the 5-shot model starting with a poor initial hypothesis, even containing non-target language words, it is able to adapt to user corrections and continue the translation. In contrast, the LoRA model, while generating a high-quality initial translation, loses its generalization capability. As a result, when the user provides new feedback, the model frequently predicts the end-of-sentence (EOS) token with the highest probability, leading to worst WSR.

6 Conclusions and future work

In this study, we implemented a RL algorithm on the LoRA method to train various versions of the BLOOM LLM. The primary objective was to compare the results obtained with the base model using few-shot prompting in an IMT setting, aiming to minimize the effort required by human users to correct the generated translations.

Using the LoRA method has significantly improved the quality of the translations produced by the models, demonstrating its capability to fine-tune model weights for specific tasks. However, in the context of IMT, although LoRA models start with a better hypothesis, they struggle with generalization and adapting to user modifications. Despite enhancements in translation generation, LoRA models often continue with an end-of-sentence token when forced to use a less probable word. This forces users to input the entire translation manually, disrupting the interactive environment intended to simplify the translation process.

Although LLM are increasingly being utilized across many fields to optimize and enhance results, the best outcomes in the IMT field are still achieved with models trained explicitly for the domain, even if they are considerably smaller in size. In this context, the application of RL has not shown significant improvements compared with the standard usage of LoRA. Future work could explore employing RL in other domains or conducting full model fine-tuning instead of using LoRA. Additionally, investigating other PEFT methods in IMT environments could be valuable in determining if they offer improvements without compromising the model's generalization capabilities.

7 Limitations

When utilizing LLM such as the BLOOM model, which we have employed for this study, we are constrained by the substantial memory requirements necessary for their use, as well as the computational time required for both fine-tuning and inference. By employing the LoRA technique, we are able to leverage different versions of the same model without significantly increasing the memory footprint. This is achieved by alternating between LoRA files, which only require a few megabytes of storage. Nevertheless, it remains necessary to load the base LLM itself, which in and of itself poses a challenge for many users.

Acknowledgements

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Vicent Alabau, Ragnar Bonk, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Jesús González-Rubio, Philipp Koehn, Luis Leiva, Bartolomé Mesa-Lao, et al. 2013. Casmacat: An open source workbench for advanced computer aided translation. *The Prague Bulletin of Mathematical Linguistics*, 100(1):101–112.

Vicent Alabau, Daniel Ortiz-Martínez, Alberto Sanchis, and Francisco Casacuberta. 2010. Multimodal interactive machine translation. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*, pages 1–4.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*.

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and Juan-Miguel Vilar. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Ona De Gibert, Graeme Nail, Nikolay Arefyev, Marta Bañón, Jelmer Van Der Linde, Shaoxiong Ji, Jaume Zaragoza-Bernabeu, Mikko Aulamo, Gema Ramírez-Sánchez, Andrey Kutuzov, et al. 2024. A new massive multilingual dataset for high-performance language technologies. *arXiv preprint arXiv:2403.14009*.

Miguel Domingo, Álvaro Peris, and Francisco Casacuberta. 2017. Segment-based interactive-predictive machine translation. *Machine Translation*, 31(4):163–185.

Marcello Federico, Nicola Bertoldi, Mauro Cettolo, Matteo Negri, Marco Turchi, Marco Trombetti, Alessandro Cattelan, Antonio Farina, Domenico Lupinetti, Andrea Martines, Alberto Massidda, Holger Schwenk, Loïc Barrault, Frédéric Blain, Philipp Koehn, Christian Buck, and Ulrich Germann. 2014. The matecat tool. In *Proceedings of the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 129–132, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

George Foster, Pierre Isabelle, and Pierre Plamondon. 1997. Target-text mediated interactive machine translation. *Machine Translation*, 12(1):175–194.

Nico Herbig, Tim Düwel, Santanu Pal, Kalliopi Meladaki, Mahsa Monshizadeh, Antonio Krüger, and Josef van Genabith. 2020. MMPE: A Multi-Modal Interface for Post-Editing Machine Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1691–1702, Online. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Guoping Huang, Lemao Liu, Xing Wang, Longyue Wang, Huayang Li, Zhaopeng Tu, Chengyan Huang, and Shuming Shi. 2021. Transmart: A practical interactive machine translation system. *arXiv preprint arXiv:2105.13072*.

Nam Hyeon-Woo, Moon Ye-Bin, and Tae-Hyun Oh. 2021. Fedpara: Low-rank hadamard product for communication-efficient federated learning. *arXiv preprint arXiv:2108.06098*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation summit*, volume 5, pages 79–86. Citeseer.

Julia Kreutzer, Shahram Khadivi, Evgeny Matusov, and Stefan Riezler. 2018. Can neural machine translation be improved with user feedback? *arXiv preprint arXiv:1804.05958*.

Tsz Kin Lam, Shigehiko Schamoni, and Stefan Riezler. 2019. Interactive-predictive neural machine translation through reinforcement and imitation. *arXiv preprint arXiv:1907.02326*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. Gpt understands, too. *AI Open*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. ACL.

Álvaro Peris and Francisco Casacuberta. 2018. Active learning for interactive neural machine translation of data streams. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 151–160, Brussels, Belgium. ACL.

Álvaro Peris and Francisco Casacuberta. 2019. Online learning for effort reduction in interactive neural machine translation. *Computer Speech & Language*, 58:98–126.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation*, pages 186–191.

Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. 2023. Controlling text-to-image diffusion by orthogonal finetuning. *Advances in Neural Information Processing Systems*, 36:79320–79362.

Germán Sanchis-Trilles, Vicent Alabau, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Ulrich Germann, Jesús González-Rubio, Robin Hill, Philipp Koehn, et al. 2014. Interactive translation prediction versus conventional post-editing in practice: a study with the casmacat workbench. *Machine Translation*, 28(3-4):217–235.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. AMTA.

Lucia Specia, Kashif Shah, José GC De Souza, and Trevor Cohn. 2013. Quest-a translation quality estimation framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84, Sofia, Bulgaria. ACL.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Jesús Tomás and Francisco Casacuberta. 2006. Statistical phrase-based models for interactive computer-assisted translation. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 835–841, Sydney, Australia. ACL.

Antonio Toral. 2020. Reassessing claims of human parity and super-human performance in machine translation at wmt 2019. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 185–194, Lisboa, Portugal. EAMT.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Qian Wang, Jiajun Zhang, Lemao Liu, Guoping Huang, and Chengqing Zong. 2020. Touch editing: A flexible one-time interaction approach for translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 1–11.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. 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.

Wojciech Zaremba and Ilya Sutskever. 2015. Reinforcement learning neural turing machines-revised. *arXiv preprint arXiv:1505.00521*.