Unsupervised multimodal machine translation using visual signals as rewards for reinforcement learning

Gargeya Sharma
School of Science & Engineering (EECS)
Queen Mary University of London
London, United Kingdom
gargeya.sharma@gmail.com

***Abstract*—Unsupervised machine translation, a rapidly evolving field within natural language processing, aims to develop translation models without relying on parallel corpora. This paper explores the novel approach of using visual signals as rewards within a reinforcement learning framework for unsupervised multimodal machine translation. By leveraging visual cues, the proposed method avoids the need for paired source and target language sentences, making it a more scalable and cost-effective solution. The research employs an on-policy policy gradient method in an attempt to optimize the translation policy and utilizes adversarial training techniques to improve the model's performance. Evaluation metrics such as BLEU and METEOR are utilized to assess the quality of the generated translations. The experimental setup includes the integration of visual reasoning scores from a multimodal transformer model, acting as rewards for the reinforcement learning agent. The baseline has been set with the pre-trained Large Language Model before fine-tuning. This work contributes to the diversity of approaches for unsupervised machine translation and provides direction for the research community to further explore this multimodal setup to facilitate cross-lingual communication and understanding.**

***Index Terms*—Unsupervised Learning, Machine Translation (MT), Reinforcement Learning (RL), Natural language Processing (NLP), Visual Signals, Policy Gradient**

I. Introduction

In today's interconnected global landscape, linguistic diversity remains a significant challenge. Language is a cornerstone of human cognition, enabling intricate expression and communication. Despite the digital age's connectivity, linguistic barriers persist, often hindering mutual understanding. Modern machine learning and artificial intelligence have been able to overcome several of these hurdles by developing methods that automatically translate a text written in one language to another, also termed Machine Translation (MT). Traditional MT approaches rely heavily on parallel data for supervised learning, making data collection, and its expert curation a challenging and time-consuming task. Due to such processes, the availability of highly curated data becomes a bottleneck in further widening the scope of machine translation in order to include languages that are not readily available in correspondence to the available data. This limitation among others gives rise to the field of unsupervised machine translation (UMT), a subfield of Natural Language Processing (NLP) that doesn't depend on parallel corpora of bi/multilingual source and target language sentences; in order to learn the inter-syntactic and semantic mapping between them. For several years, researchers have been exploring alternative methods to eradicate the use of target language during the training of MT models. More on this is discussed in the later section. A pivotal approach, which ignited the idea of this paper and heavily influenced the direction of the author's research, is to leverage unsupervised rewards in Machine Translation via reinforcement learning by Julia Ive et.al [1] Their paper demonstrated the potential of the unsupervised reward function using soft actor-critic (SAC) algorithm (a type of reinforcement learning's adversarial training setup) for the cases when they had to choose between possible translations for an ambiguous word (i.e., better exploration of the search space).

Reinforcement learning (RL), a subfield of machine learning, plays a crucial role in this paper. In general, RL focuses on training agents to make sequential decisions in an environment to maximize cumulative rewards. In the context of unsupervised machine translation, RL algorithms, such as Proximal Policy Optimization (PPO), are employed to train translation models. These models, often based on neural network architectures, take a source sentence as input and generate a target sentence as output. By interacting with an environment and receiving either supervised or unsupervised rewards, the translation model learns to improve the quality of its generated translations.

This paper introduces a novel approach to train a UMT model via RL using unsupervised multimodal rewards. The author is utilizing the power of trained and pretrained transformers to build an RL pipeline where the visual signals serve as the basis for rewards and are calculated using a trained vision-text-based multimodal transformer. Those rewards facilitate the fine-tuning of a pretrained Large Language Model (LLM) through additional context available on the data during training, to perform unsupervised machine translation. Training an MT model using PPO itself is a challenging task due to the erratic nature of RL, There is hardly any literature available on machine translation tasks using PPO, so the conclusive approach mentioned in this paper was finalized after countless experiments and training. The finalized unsupervised MT model performs German to English translations on a competitive level to the baseline, while their difference is statically not significant with the p-value of 0.01 on a Permutation test.

In addition to unsupervised training, the author has also performed supervised training to confirm that the PPO system works just fine and that the model performance is not being manipulated by the training framework. After any training, evaluation metrics play a crucial role in assessing the quality and effectiveness of any model, especially an unsupervised one. Commonly used metrics, such as BLEU (Bilingual Evaluation Understudy) and METEOR (Metric for Evaluation of Translation with Explicit Ordering) are being used in this paper to evaluate the performance of the model. These metrics compare the generated translations against reference translations and measure their similarity, precision, recall, and alignment. These also provide quantitative measures to evaluate the translation model's progress and facilitate comparisons between different approaches.

This paper presents the literature review, methodologies, experimental setup, results and conclusion of the author's research, highlighting the challenges faced, the approaches that were undertaken, and the potential of unsupervised machine translation using visual signals as rewards for reinforcement learning. The hypothesis is that visual cues can provide a reliable form of 'universal language' that can help bridge the gap between different languages in the absence of parallel text. The author aims to contribute to advancing the understanding and capabilities of this research topic and broaden the field of machine translation to facilitate cross-lingual communication with ease.

II. Literature Review

A. *Reinforcement learning in machine translation*

Everything changed when OpenAI released their paper introducing ChatGPT, With the latest version (2023) of this paper "Summary of ChatGPT/GPT-4 Research and Perspective Towards the Future of Large Language Models" [2], their findings reveal a significant and increasing interest in ChatGPT/GPT-4 research, predominantly centered on direct natural language processing applications. This step boosted the relationship between NLP and RL. Before that in terms of MT, RL has been used to make training and testing better by directly focusing on the main goal as shown by Yu et al., (2017) [3]; Ranzato et al., (2015) [4]; Bahdanau et al., (2017) [5]. But, most of the time, they've used the REINFORCE (Williams, 1992) [6] algorithm and its variants (Ranzato et al., 2015 [4]; Kreutzer et al., 2018 [7]). These methods are basic and can handle the many options in language, but they don't give many rewards.

To make the model steadier, Actor-Critic (AC) models check the reward at every step and use the Critic model to decide what to do next (Konda and Tsitsiklis, 2000) [8]. This method has been tried in MT too as shown by Bahdanau et al., (2017) [5] and He et al., (2017) [9]. But, more complex AC models with Q-Learning aren't used much for creating language because it's tough to guess the Q-function with so many choices. Choshen et al., (2020) [10] show that having too many choices is a big challenge for using RL in making text. It's crucial to start the agent's training close to the real outcome for RL to work well.

Additionally, The use of reinforcement learning in machine translation has been explored by several researchers. "Deep Reinforcement Learning for Dialogue Generation" by Li, J. et al. (2016) [11] proposes a framework that uses reinforcement learning to generate more diverse and interesting responses in dialogue systems. This work demonstrates the potential of RL in improving the quality of generated text in various NLP tasks. Nguyen et.al. (2017) [12] proposed that MT is a natural candidate problem to be solved with reinforcement learning from human feedback, having users provide quick, dirty ratings on candidate translations to guide a system to improve. [12] introduces a reinforcement learning algorithm that improves MT from simulated human feedback using the combination of advantage actor-critic (A2C) with an attention-based neural encoder-decoder system. Their work resulted in better performance on problems with large action space and delayed rewards, effective optimization and increased robustness to skewed, high variance and granular feedback just like how humans behave.

'MAD' by Domenic Donato et.al., (2022) [13] not only utilizes the training stability and efficient performance backed by PPO for fine-tuning LLMs for machine translation as a reference but also outperforms it with their approach. Through the paper "Proximal Policy Optimization Algorithms" by OpenAI, (2017) [14], it can be concluded that PPO offers stable training in reinforcement learning by limiting policy updates, ensuring consistent performance. Its sample efficiency makes it ideal for tasks with limited data. PPO effectively handles sparse rewards, common in NLP, and balances exploration and exploitation. Its flexibility allows adaptability across diverse tasks without extensive hyperparameter tuning.

The author realized that there is a significant gap in research while solving unsupervised machine learning problems with PPO, especially UMT. This gap provided a great opportunity and motivation for the author to move forward with the novel research on unsupervised multimodal machine translation using PPO. The later section will shed some light on why multimodal learning was opted to tackle this task.

B. *Multimodal learning*

Multimodal deep learning focuses on developing and training models capable of handling, processing, and connecting data from diverse modalities Baltrušaitis et al., (2017) [15]. Several reviews highlight the advantages of multimodal methods over unimodal ones for tasks like retrieval, matching, and categorization (Atrey, P.K. et.al. 2010 [16], Bhatt, C.A. and Kankanhalli, M.S. (2010) [17]). [15] emphasize the inherently multimodal nature of our world, where experiences span across visual, auditory, tactile, olfactory, and gustatory senses. Their comprehensive survey underscores the significance of various modes of sensory data in multimodal machine learning, which seeks to interpret these diverse signals cohesively.

A significant stride in this direction is the work by Jiang et al., (2021) [18], which introduces a novel multimodal deep learning architecture, TechDoc. This architecture processes technical documents that often contain both text and images and achieves superior classification accuracy for technical documents based on the hierarchical International Patent Classification system.

Even for the financial sector, the latest contribution by Cho et al.(2023) [19], proposes an intelligent document processing framework for the financial sector. This framework combines traditional robotic process automation (RPA) with a pre-trained deep learning model to process real-world financial document images. The proposed solution demonstrates the efficacy of a multimodal approach in understanding financial documents, even with limited training data, and can handle multilingual documents with ease.

In conclusion, multimodal machine learning is proving to be indispensable in tackling complex challenges across various domains, showcasing its versatility and high potential.

C. Unsupervised machine translation

In the field of machine translation, unsupervised learning has emerged as a promising approach to train models without the need for parallel corpora. In the paper ”Unsupervised Neural Machine Translation” by Artetxe et al. (2018) [20] the authors proposed an initial unsupervised training step that uses a shared encoder and a shared decoder for both languages which was competitive with supervised methods on low-resource language pairs, followed by a language modelling step that refines the model using monolingual corpora. Lample et al. (2018) [21] also worked with monolingual data and proposed a method for UMT that uses a shared latent space and a cycle consistency loss. Their work has been widely cited and has inspired subsequent research in the area.

A few major inspirations for this research and methodology are derived from these couple of papers: The paper ”SURF: Semantic-level Unsupervised Reward Function for Machine Translation” by Julia Ive et.al.(2022) [22] presents a novel approach that leverages semantic-level rewards for reinforcement learning in machine translation. The authors propose a reward function that is based on semantic similarity between the source and target sentences, which is computed using pre-trained language models. This approach allows the model to focus on preserving the meaning of the sentence during translation, rather than just matching the exact words or phrases. As mentioned in the Introduction of the paper [1] ”Exploring supervised and Unsupervised rewards for machine translation” (2021), the authors investigate the use of both supervised and unsupervised rewards in reinforcement learning for machine translation. They found that combining both types of rewards can lead to better performance than using either type alone.

D. Multimodal Transformers

The field of machine translation has also seen the integration of multi-modal learning, where models leverage information from multiple modalities, such as text and images, to improve performance. The paper ”ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks” (2019) [23] presents a model that learns joint representations of image and text data. The authors demonstrate that this approach can significantly improve performance on several vision-and-language tasks.

Along the same lines, the novelty of this paper arises by utilizing visual reasoning as the unsupervised multimodal reward criteria for UMT. To design the reward, the author is using The Vision-and-Language Transformer (ViLT) model, introduced in the paper ”ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision” (2021) [24]. The ViLT model leverages visual cues along with the text to provide a score that helps bridge the gap between different languages in the absence of parallel text. [24] demonstrate that this approach can achieve competitive performance on several benchmarks, while being simpler and more efficient than predecessor methods (ViLBERT [23] (2019), VisualBERT (2019) [25] etc.), thereby addressing another key challenge in unsupervised MT.

The majority of multimodal transformer applications revolve around solving NLP tasks using images or vice versa, ”AI Ekphrasis: Multi-Modal Learning with Foundation Models for Fine-Grained Poetry Retrieval” (2022) [26] paper discusses a deep learning approach for the automatic retrieval of poetry suitable to the input images. The approach takes advantage of strong pre-training of the CLIP model (OpenAI, 2021) [27] and overcomes its limitations by introducing shared attention parameters to better model the fine-grained relationship between both modalities.

All these research evidences clearly points towards a high potential direction of utilizing multimodal data and learning to enhance the rate of success with highly complex tasks, and unsupervised machine translation is clearly one of them.

The author began the research with a powerful multitasking LLM: T5 (Text-to-Text Transfer Transformer) (2020) [28], a state-of-the-art model for NLP tasks, that is pre-trained on a large corpus of text and then fine-tuned for specific tasks. Its ability to handle different NLP tasks by converting them into a text generation problem makes it a versatile tool for this research. During the experimentation, mentioned in the section below, T5 gets replaced by another LLM: Opus (2020) [29], a model that is trained on a collection of translated texts from the web, providing a rich resource for training and testing translation models. Its extensive multilingual and multi-domain dataset capabilities allow us to train our model on a diverse range of texts, thereby enhancing the model's robustness and generalization capabilities.

III. Methodology

A. Neural Machine Translation (NMT)

Neural Machine Translation (NMT) commonly employs a sequence-to-sequence (seq2seq) structure (Sutskever et al., (2014) [30]; [5]). In this setup, a source sentence $x = (x_1, x_2, \ldots, x_n)$ is transformed by the encoder into a set of hidden states. During each decoding phase t, a target word y_t is produced based on $p(y_t|y_{<t}, x)$, which is conditioned on the input sequence x and the decoded sequence $y_{<t} = (y_1, \ldots, y_{t-1})$ up to the t-th step. Given a corpus of source and target sentence pairs $\{x_i, y_i\}_{i=1}^{N}$, the training goal, or maximum likelihood estimation (MLE), is expressed as (equation 1):

$$L_{MLE} = -\sum_{i=1}^{N}\sum_{t=1}^{T} p(y_t^i|y_1^i, \ldots, y_{t-1}^i, x_i)$$

(1)

B. Reinforcement Learning for NMT

In the Reinforcement Learning (RL) context, NMT is viewed as a sequential decision-making task. Here, the state is represented by previously generated words $y_{<t}$ and the action is the upcoming word to be produced. Given state s_t, an agent selects an action a_t (equivalent to y_t in seq2seq) based on a policy π_θ and receives a reward r_t for that action, which can be determined using metrics like BLEU.

The RL training's objective is to optimize the expected reward (equation 2):

$$L_{RL} = E_{a_1,\ldots,a_T \sim \pi_\theta(a_1,\ldots,a_T)}[r(a_1, \ldots, a_T)]$$

(2)

Under policy π, the values of the state-action pair $Q(s_t, y_t)$ and the state $V(s_t)$ can be defined as (equation 3):

$$Q_\pi(s_t, a_t) = E[r_t|s = s_t, a = a_t]$$
$$V_\pi(s_t) = E_{a\sim\pi(s)}[Q_\pi(s_t, a = a_t)]$$

(3)

Conceptually, the value function V gauges the potential of a model in a specific state s_t. The Q function evaluates the worth of selecting a particular action in that state.

From these definitions, an advantage function, represented as A_π, which relates V and Q, can be defined (equation 4):

$$A_\pi(s_t, a_t) = Q_\pi(s_t, a_t) - V_\pi(s_t)$$

(4)

The goal then becomes to optimize one of these objectives (equation 5):

$$\max_a A_\pi(s_t, a_t) \rightarrow \max_a Q_\pi(s_t, a_t)$$

(5)

Different RL methods employ varied strategies to find the best policy. Techniques like REINFORCE and its variant MIXER [4], which are prevalent in linguistic tasks, seek the best policy using Eq. 2 through the Policy Gradient. Actor-Critic (AC) models typically enhance the Policy Gradient models' performance by addressing Eq. 5's left side [5]. Q-learning models focus on maximizing the Q function (Eq 5, right side) to surpass both the Policy Gradient and AC models (Dai et al., 2018) [31].

Proximal Policy Optimization (PPO) is another advanced reinforcement learning algorithm that comes under the category of on-policy policy gradient methods and has gained significant popularity due to its stability and efficiency [14], this research takes advantage of that. Unlike traditional policy gradient methods that can have large policy updates, potentially leading to suboptimal policies, PPO aims to take the largest step possible while ensuring the new policy doesn't stray too far from the old policy. This is achieved by adding a constraint to the policy update. There are two primary variants of PPO: PPO-Penalty and PPO-Clip. In this paper, the words PPO-Clip and PPO are used interchangeably because that is the primary variant used by both OpenAI and the author.

The objective function for PPO is given by (equation 6):

$$L^{CLIP}(\theta) = \hat{E}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)]$$

(6)

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ is the ratio of the new policy to the old policy, and $\hat{A}_t$ is an estimator of the advantage function at time t. The clip function ensures that the ratio $r_t(\theta)$ remains within a specified range, defined by ϵ, thus preventing overly large policy updates.

In essence, PPO strikes a balance between making significant policy updates to improve performance and ensuring these updates don't destabilize the learning process. This balance has made PPO a preferred choice for many reinforcement learning tasks, including some NMT challenges.

C. Reward Formulation

The heart of any reinforcement learning system is its rewards. In this paper, we utilize two different reward functions for two models (unsupervised and supervised). The author used **ViLT** for the unsupervised approach, It is a multimodal transformer model [24] that is designed to handle a variety of tasks such as visual question answering, natural language for visual reasoning, image-to-text and text-to-image retrieval. The major reason for involving ViIT is its unified embedding space. ViLT processes both images and text to represent them in a shared embedding space. This means it transforms both visual and textual data into a format where they can be compared, related, or jointly processed; which further allows a transformer architecture to capture intricate relationships and dependencies between visual and textual elements. For the main aim of this paper (unsupervised model), Images and texts are used in unison to generate rewards via ViLT transformer with a classifier head on top for image-to-text and text-to-image retrieval. This particular functionality produces classification logit scores that validate how well the text and image align with each other. The range of this score lies between (-inf, inf) where the higher the score, the better the alignment of the text with the image and vice versa, the lower the score, the poorer the reasoning. Hugging face, an open-source library mentioned in the later section, implemented ViLT and its functionalities including the one utilized in this paper and made it publicly available through their 'transformers' python package. There are 5 steps to retrieve this ViLT score using hugging face implementation:

1. Make a processor object by loading the ViltProcessor class with the same pretrained configuration as used by the model in step 2.

2. Make a model object by loading the ViltforImageAndTextRetrieval class with a certain pretrained configuration (”dandelin/vilt-b32-finetuned-coco” in our case).

3. Pass the input image and text into the processor with the required output configuration such as max_length, truncation, return type of the output etc.

4. Get the output from step 3 and pass it as input to the model

5. The output from the model comprises various elements depending on the model's configuration and inputs, the scores are the 'logits' values included in the output.

The raw score is one of the ways to set rewards for the PPO, but section 4 (Experimental setup), will determine the paper's best practice along with other experimental approaches.

Supervised Reward: To prove the proper functioning of the PPO pipeline, another fine-tuning was performed on the pre-trained model as a separate system. This time, the reward function utilized the target language text descriptions to calculate a supervised reward. This paper utilizes one of the best practices of the community and sets a cumulative BLEU score between the predicted text and the target text as the reward. For more clarity think of it this way, there are multiple evaluation pairs for each target and predicted text where the target text remains the same in every evaluation, but the predicted text is dissected into a cumulative combination of words ([first_word, first_word second_word, first_word, second_word, third_word ..., entire predicted text]). The BLEU score generated from each pair is summed up to get an aggregate BLEU score for the specific pair of sentences, which in this paper acts as the supervised reward.

D. Implementation Details

Software and Tools:

- **Jupyter Notebook**: Used for interactive development and real-time visualization.
- **Python**: The project was implemented in Python 3.9.

Libraries:

- **PyTorch**: The main deep learning framework.
- **Transformers**: From Hugging Face, used for accessing pre-trained models and tokenization.
- **TRL**: Integrated reinforcement learning with transformer architectures.
- **NLTK**: Used for calculating the METEOR score.
- **SacreBLEU**: Evaluated machine translations quality.
- **WandB**: Employed for experiment tracking, optimization, and visualization.

Hardware: The Jupyter notebook was accessed via QMUL's resources. System configurations: **CPU:** Intel(R) Xeon(R) Gold 5222 @ 3.80GHz, **RAM:** 19GB, and a 24 GB NVIDIA Quadro RTX 6000 **GPU**. Training averaged 6 hours, varying with batch size choices.

IV. Experimental Setup

A. Dataset

The author utilized the Multi30K dataset [32], comprising images with descriptions in multiple languages. This study focused on English and German descriptions. The textual data was sourced from the 'multi30k/dataset' GitHub repository, while the images were obtained from the University of Illinois. The dataset includes 29,000 training, 1,010 validation, and 1,000 test instances. Evaluation employed **flickr2016 (2016)**, **flickr2017 (2017)**, **flickr2018 (2018)**, and **coco2017 (COCO)** test sets. **2016** is **in-domain**, derived from the training pool, while **2017**, **2018**, and **COCO** are **out-of-domain**. Images of this dataset need to be requested through this link.

Preprocessing: The author created a function to load image names, German, and English descriptions from separate text files. This function accepts three parameters: input language, output language, and split, which determines the dataset used. The data is organized into a list of dictionaries, each representing a dataset instance. These dictionaries are then updated with ”input_ids” and ”query” key-value pairs for tokenization purposes. The RL model is instantiated using the 'PPOTrainer' class from the 'trl' library, which requires data in the 'torch.utils.data.Dataset' format. The final step converts the list of dictionaries to this required format.

B. Model Selection

The author experimented with unsupervised machine translation using visual rewards in reinforcement learning. Initially, traditional models like T5, BART, and GPT-2 were assessed for machine translation using the Multi30K dataset. The T5 model's performance, gauged by the BLEU score, was satisfactory. The next step integrated T5 with the ViLT model for reward generation, combining predictions and visual reasoning. However, a misalignment arose as T5's German outputs didn't match ViLT's English input for visual reasoning. To resolve this, the author researched more and switched to the Opus model, which supported German(de)-to-English(en) translation. Both T5 and Opus had comparable BLEU scores (34.368 and 34.687 respectively), ensuring consistent performance and successful unsupervised machine translation integration.

C. Working of the System & Training

In section 4. B, Opus was finalized as the LLM model to perform this research. In this section, the entire working pipeline of the system as well as model training is explained sequentially with the help of Fig.1. to provide a detailed understanding of what is happening and how. The author developed the figure in such a way that makes the implementation self-explanatory in the form of a visual algorithm.

In Fig. 1. each arrow holds the sequence number and direction of the data from one component to another. Here is what they mean:

1) The model store, an open-source model repository by Hugging Face, provides the necessary models (Opus

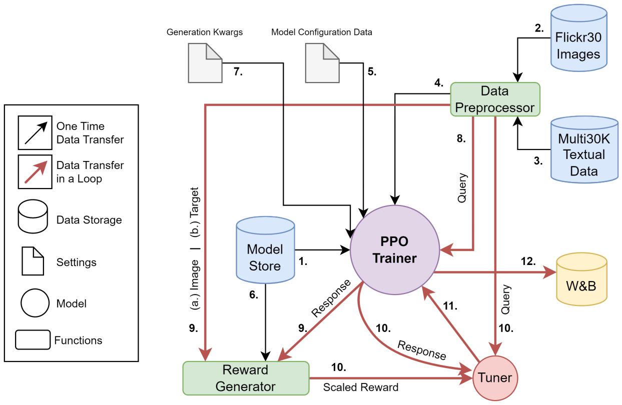


Fig. 1. The system's end-to-end operation is visually represented in the paper's figure, serving as a high-level algorithm guide. The steps are: 1. Load the pretrained opus model and tokenizer into PPOTrainer, 2-3. Preprocess Multi30K data, 4. Input data into PPOTrainer, 5. Set training hyperparameters, 6. Establish the ViLT reward function, 7. Configure PPO settings, 8. Process a batch of queries, 9. Pass image and target text into the reward function with model predictions, 10. Calculate rewards and adjust the model, 11. Optimize and fine-tune the model, 12. Save logs on W&B for real-time monitoring.

and PPOTrainer) and tokenizers through libraries like 'transformers' and 'trl'. A pretrained opus model copy is created for calculating the Kullback-Leibler (KL) divergence during training.

2) Raw images are unzipped and their directory paths are set for preprocessing, as detailed in section 4. A.
3) The GitHub repository for Multi30K textual data is accessed, and its files are processed for PPOTrainer, as per Section 4. A.
4) The preprocessed dataset, Opus models, and tokenizer are integrated with the PPOTrainer class.
5) Model configuration data, including hyperparameters and logging details, are passed before instantiating the PPOTrainer object, allowing for training and real-time monitoring on Weights & Biases.
6) A reward function for the unsupervised approach is created using the pretrained ViltProcessor and ViLT's model, as described in section 3.C.
7) Parameters are set to control the output type from the LLM, including input length, token sampling, padding, and output translation length.
8) Red arrows indicate continuous data flow, signifying ongoing training, while black arrows represent one-time transfers. Training starts with the dataset providing batches to the model, which then produces response tensors.
9) The core of the training involves detokenizing the response tokens, removing padding, and passing them to the reward function with either image names (unsupervised) or target language descriptions (supervised).
10) The batch of rewards is used alongside queries and response tensors to fine-tune the model based on the learning rate set during Step 5.
11) The tuner adjusts the model parameters to optimize task performance and increase the reward for its predictions, either unsupervised or supervised.
12) Post fine-tuning, model behavior, rewards, entropy, KL divergence, and other metrics are logged on weights & biases for this specific training task.

The cycle from step 8 to step 12 repeats itself until all the batches are exhausted, The model has seen the entire training dataset while performing the number of steps mentioned in the model's configuration (step 5) and the training has run for the specified number of epochs. In this research, the number of epochs is set to be 1 in all experimental cases.

D. *Experiments*

The author put a lot of effort and time into getting this system up and running with significant results to share with the community. This section contains all the wrong turns that were made during the research and how certain settings resulted in bizarre and unethical results that should be avoided while fine-tuning with language models. In the author's opinion, this section provides a lot of guidance to the community to avoid the same mistakes as the author's and save themselves plenty of time and failures.

System setup:

During the early model training phase, several challenges arose. Initially, two different opus models were used within the training loop. The primary model generated response tokens, while another pretrained opus model, loaded using Hugging Face's pipeline, provided English translations for the reward function. This approach prevented the training model from using its predictions for rewards. The author later rectified this by using the decoder to obtain detokenized English translations from response tokens.

Furthermore, determining the optimal 'max_length' for tokenized queries and deciding on padding took multiple trials. Excessive padding led to unnecessary word generation and nonsensical repetitions in predictions. After various tests, padding was disabled, truncation was enabled, and max length was set to 25.

It's also vital to exclude special or padding tokens when detokenizing response tensors. Overlooking this introduced noise in predictions, affecting reward calculations. To refine predictions, the author used the arguments skip_special_tokens=True and clean_up_tokenization_space=True in the tokenizer.decode function. Additionally, generation kwargs were adjusted as per step 7. Initial settings caused errors due to tensor size mismatch with ViLT's maximum input size and autorange ValueError on getting nan values during logging in step 12. Resolutions included adjusting the ViltProcessor object's max_length to 40 and modifying the minimum length parameter in generation kwargs to 2 in step 7. To enhance generation quality by eradicating token repetition and gibberish translations, the max new tokens value was reduced from 99 to 13.

Hyperparameters and Rewards:

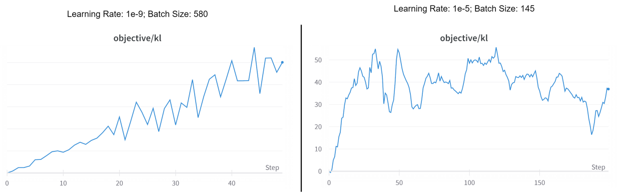


Fig. 2. KL Divergence curve during training with different learning rates and batch sizes. The chart above shows that there is a steady and small increase in KL divergence, which is ideal for the system, whereas the chart below shows an explosion in KL divergence value by crossing the threshold value of 10 and damaging the learning of the model.

The biggest leap in improving the quality of the training and stabilizing the performance of both unsupervised and supervised models; is related to setting the appropriate learning rate and batch size in the model configuration settings (Step 5.). Fig 2. shows a performance comparison of how learning rate and batch size affected the training curve in the form of avoiding the explosion of KL divergence beyond a reasonable limit (10). Among all the combinations between various learning rates and batch sizes, the author believes that for specifically Multi30K dataset, the best values out of all the tried options are 1e-9 and 580 respectively. For more details, check out the saved experimental runs of the model on author's weight & biases profile.

Unsupervised Rewards

The reward function is pivotal in RL systems, and extensive experimentation was conducted on it. The author explored using raw logit scores from the ViLT model, discretizing these scores with various bin ranges and values, and adjusting bin numbers from 4 to 3 to 2. Discrete rewards were ineffective, leading to a shift to normalized rewards. These were bounded between +10 and -10, then min-max scaled. However, normalization within (0, 1) led to training errors such as encountered before: Value error, the autorange detect [nan, nan]. Thus, rewards were rescaled to ranges like 0 to 10, 0 to 100, -50 to 50, and others. Negative minimums aimed to reflect ViLT's visual reasoning scores. Batch normalization was also tested, with rewards scaled using different multipliers. Attempts to change rewards from linear to quadratic or cubic were unsuccessful due to gradient optimization concerns. Ultimately, rewards were globally normalized using min-max scaling between -8 and +8, then scaled to a 0-100 range.

Supervised Rewards

During supervised reward experiments, using BLEU values between target and prediction as policy rewards was ineffective. Despite the method in section 3.C being standard, the author noticed a direct relationship between run-time rewards and loss graphs, contrary to expectations. Experimentally, the reward function was inverted to enforce the expected relationship. However, this approach backfired in an unethical, unexpected and disturbing way, Although now the behavior was intuitive, the increase in KL divergence resulted in the presence of pornographic words in the prediction descriptions. Ultimately, this method was abandoned, and the model was fine-tuned using the standard reward function (section 3.C) but with a reduced learning rate (from 1e-8 to 1e-9).

E. Evalutaion

A standard set of MT evaluations metrics are used: BLEU (Papineni et al., 2001) [33] and METEOR (Denkowski & Lavie, 2014) [34] to evaluate and compare the performance of pretrained, unsupervised and supervised models. Additionally, significance testing is performed on BLEU scores generated for in-domain **2016** testset via all the models mentioned above to compare the differences between pretrained and supervised, and pretrained and unsupervised.

V. Results

Here's a more concise version of the provided content:

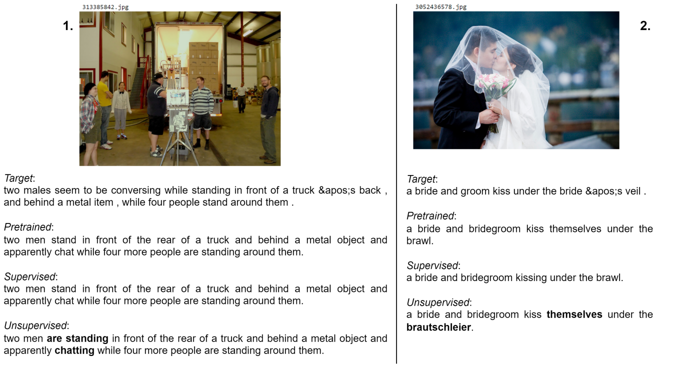
Experiments led to optimal settings for both supervised and unsupervised models, with results validating the effective functioning of the PPO policy, as seen in the subsequent sections.

The BLEU score function is sourced from 'sacrebleu', while the METEOR score function is custom-built using 'nltk', computing the mean score across the dataset. Both BLEU (Table 1) and METEOR (Table 2) scores evaluate the models (pretrained, supervised, and unsupervised) across five datasets (validation, 2016, 2017, 2018, 2017 COCO) as detailed in section 4.A. The pretrained Opus model utilizes MarianMTModel and MarianTokonizer for the model and tokenizer, which are loaded with trained parameters for evaluation. The validation dataset, untouched during training, serves as an additional in-domain testset. The supervised model's performance is first compared to the baseline to validate the PPO pipeline and set a foundation for discussing the unsupervised model's results.

Comparison with the baseline

Supervised Model

As seen in Table 1, supervised is out-performing the others on in-domain validation and 2016 testset (+0.01 BLEU and +0.43 BLEU respectively), while on out-of-domain 2017, 2018 and 2017 COCO testsets (-0.03 BLEU, -0.05 BLEU and



Target:
two males seem to be conversing while standing in front of a truck &apos;s back , and behind a metal item , while four people stand around them .

Pretrained:
two men stand in front of the rear of a truck and behind a metal object and apparently chat while four more people are standing around them.

Supervised:
two men stand in front of the rear of a truck and behind a metal object and apparently chat while four more people are standing around them.

Unsupervised:
two men **are standing** in front of the rear of a truck and behind a metal object and apparently **chatting** while four more people are standing around them.

Target:
a bride and groom kiss under the bride &apos;s veil .

Pretrained:
a bride and bridegroom kiss themselves under the brawl.

Supervised:
a bride and bridegroom kissing under the brawl.

Unsupervised:
a bride and bridegroom kiss **themselves** under the **brautschleier**.

Fig. 3. Two examples showcasing target description, their corresponding images and predicted descriptions from pretrained (baseline), supervised and unsupervised models in order to perform qualitative analysis of the proposed approach (unsupervised)

TABLE I
PERFORMANCE EVALUATION WITH BLEU SCORE ON VARIOUS TESTSETS

Evaluation	**BLEU**				
Models	Validation	Test 2016 Flickr	Test 2017 Flickr	Test 2018 Flickr	Test 2017 mscoco
Pretrained	35.39	34.98	**35.87**	**33.10**	28.26
Supervised	**35.40**	**35.41**	35.84	33.05	**28.38**
Unsupervised	34.81	34.84	35.17	32.32	27.89

TABLE II
PERFORMANCE EVALUATION WITH METEOR SCORE ON VARIOUS TESTSETS

Evaluation	**METEOR**				
Models	Validation	Test 2016 Flickr	Test 2017 Flickr	Test 2018 Flickr	Test 2017 mscoco
Pretrained	**0.5847**	0.5671	**0.5823**	0.5578	0.5326
Supervised	0.5838	**0.5680**	0.5811	**0.5593**	**0.5336**
Unsupervised	0.5805	0.5672	0.5778	0.5545	0.5317

+0.12 BLEU respectively), its performance is still evaluated better but with METEOR (Table 2)(-0.012 METEOR, +0.015 METEOR and +0.010 METEOR respectively) than BLEU. In overall, supervised Opus on Multi30K beats pretrained Opus, which is what to be expected with supervised fine-tuning. Additionally, after performing significance testing on the BLEU scores obtained from pretrained and supervised models, the value from the Permutation test: 0.0021 is less than the p-value (0.01) hence, it can be said that the difference between this pair of models is statistically significant. This result validates the optimization performed by the PPO in the author's implementation, hence, making the evaluation scores on unsupervised approach credible.

Unsupervised Model

In Table 1, the proposed approach (unsupervised model fine-tuned using visual signals) gives a tight competition to the pretrained baseline. On in-domain validation and 2016 testset, it is keeping up with the baseline with just a slight decrease of -0.58 BLEU and -0.14 BLEU respectively while, with out-of-domain 2017, 2018, 2017 COCO testset the difference is -0.70 BLEU, -0.78 BLEU and -0.37 BLEU respectively. On the other hand, with the METEOR metric (Table 2), the unsupervised model surpassed the baseline on the in-domain 2016 testset while having a tiny difference of -0.0042, -0.0045, -0.0033 and -0.0009 METEOR score on validation, 2017, 2018, and 2017 COCO testset respectively. Additionally, after performing significance testing on the BLEU scores obtained from pretrained and unsupervised models, the value from the Permutation test: 0.9332 is more than the p-value (0.01) hence, it can not be said that the difference between this pair of models is statistically significant. These results show that this approach can be further improved to surpass the baseline in no time, It points the research community towards improving the multimodal rewards by incorporating a better multimodal model than ViLT, a more refined reward system or both and so much more. The qualitative analysis shown below proves the last point.

Qualitative Analysis

In this section, two example predictions in Fig 3 illustrate the proposed model's performance. These two examples are strategically selected to show that an unsupervised model can handle both long and short descriptions. Unlike the pretrained and supervised models, the unsupervised model benefits from the associated image context. Example (1.) in Fig 3 reveals that while the pretrained and supervised models produced identical translations, the unsupervised model improved sentence grammar by using present continuous (are standing, chatting) form, likely due to the added visual context during training. In example (2.), the pretrained and supervised models misinterpreted ”bride &apos;s (bride's) veil” as ”brawl” which means something totally different from 'veil', whereas the unsupervised model retained the original German word, preserving the translation's meaning. This retention is attributed to the model's learning from visual signals, and understanding the semantic difference between ”veil” and ”brawl”.

In conclusion, while there's room for refining the integration of visual signals in reinforcement learning of LLMs, this pioneering research demonstrates the potential of using multimodal unsupervised rewards in large language models.

VI. CONCLUSION

This paper proposes a novel approach to formulate rewards for reinforcement learning, which allows unsupervised machine translation using multiple modes of data. In this research, linguistic and visual data are combined to provide different sources of context before performing the translation. This approach in unsupervised machine translation is one of its kind and opens new doors for researchers to explore more. To validate its efficacy, a supervised approach using only linguistic data was also presented, ensuring the credibility of the unsupervised model's results. The paper details extensive experimentation to determine optimal system settings. The challenges faced during this research underscore the complexity of the task as the author had to spend a lot of time researching and failing before succeeding in this uncharted territory. While the unsupervised model's scores (using BLEU and METEOR) approached the baseline, its translation quality stood out. Future research can delve deeper into refining multimodal transformers, reward formulations, and other pivotal components of the proposed system.

References

[1] Ive, J. et al. (2021) Exploring supervised and unsupervised rewards in machine translation, arXiv.org. Available at: https://arxiv.org/abs/2102.11403v1.

[2] Liu, Y. et al. (2023) Summary of CHATGPT/GPT-4 research and perspective towards the future of large language models, arXiv.org. Available at: https://doi.org/10.48550/arXiv.2304.01852.

[3] Yu, L. et al. (2017) Seqgan: Sequence generative adversarial nets with policy gradient, Proceedings of the AAAI Conference on Artificial Intelligence. Available at: https://doi.org/10.1609/aaai.v31i1.10804.

[4] Ranzato, M. et al. (2016) Sequence level training with recurrent neural networks, arXiv.org. Available at: https://arxiv.org/abs/1511.06732v7.

[5] Bahdanau, D. et al. (2017) An actor-critic algorithm for Sequence Prediction, arXiv.org. Available at: https://arxiv.org/abs/1607.07086.

[6] Ronald J Williams. 1992. Simple statistical gradientfollowing algorithms for connectionist reinforcement learning. Machine learning, 8(3-4):229–256.

[7] Kreutzer, J., Uyheng, J. and Riezler, S. (2018) 'Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning', Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) [Preprint]. doi:10.18653/v1/p18-1165.

[8] Konda, V.R. and Tsitsiklis, J.N. (1999) Actor-Critic Algorithms. NIPS Proceedings, 13, 1008-1014.

[9] He, D. et al. (2017) Decoding with value networks for Neural Machine Translation, Advances in Neural Information Processing Systems. Available at: https://dl.acm.org/doi/abs/10.5555/3294771.3294788.

[10] Choshen, L. et al. (2020) On the weaknesses of reinforcement learning for neural machine translation, arXiv.org. Available at: https://arxiv.org/abs/1907.01752v4.

[11] Li, J. et al. (2016) Deep Reinforcement Learning for Dialogue Generation, arXiv.org. Available at: https://arxiv.org/abs/1606.01541.

[12] Nguyen, K., Daumé III, H. and Boyd-Graber, J. (2017) 'Reinforcement learning for bandit neural machine translation with simulated human feedback', Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing [Preprint]. doi:10.18653/v1/d17-1153.

[13] Donato, D. et al. (2022) Mad for robust reinforcement learning in machine translation, arXiv.org. Available at: https://arxiv.org/abs/2207.08583.

[14] Schulman, J. et al. (2017) Proximal policy optimization algorithms, arXiv.org. Available at: https://arxiv.org/abs/1707.06347.

[15] Baltrušaitis, T., Ahuja, C. and Morency, L.-P. (2017) Multimodal Machine Learning: A Survey and taxonomy, arXiv.org. Available at: https://arxiv.org/abs/1705.09406v2.

[16] Atrey, P.K. et al. (2010) Multimodal Fusion for Multimedia Analysis: A survey - multimedia systems, SpringerLink. Available at: https://link.springer.com/article/10.1007/s00530-010-0182-0.

[17] Bhatt, C.A. and Kankanhalli, M.S. (2010) 'Multimedia Data Mining: State of the art and Challenges', Multimedia Tools and Applications, 51(1), pp. 35–76. doi:10.1007/s11042-010-0645-5.

[18] Jiang, S. et al. (2022) Deep Learning for Technical Document Classification, arXiv.org. Available at: https://arxiv.org/abs/2106.14269.

[19] Cho, S. et al. (2023) A framework for understanding unstructured financial documents using RPA and Multimodal Approach, MDPI. Available at: https://doi.org/10.3390/electronics12040939.

[20] Artetxe, M. et al. (2018) Unsupervised neural machine translation, arXiv.org. Available at: https://arxiv.org/abs/1710.11041.

[21] Lample, G. et al. (2018) Unsupervised machine translation using monolingual corpora only, arXiv.org. Available at: https://arxiv.org/abs/1711.00043.

[22] Anuchitanukul, A. and Ive, J. (2022) 'Surf: Semantic-level unsupervised reward function for machine translation', Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies [Preprint]. doi:10.18653/v1/2022.naacl-main.334.

[23] Lu, J. et al. (2019) Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks, arXiv.org. Available at: https://arxiv.org/abs/1908.02265.

[24] Kim, W., Son, B. and Kim, I. (2021) Vilt: Vision-and-language transformer without convolution or region supervision, arXiv.org. Available at: https://arxiv.org/abs/2102.03334.

[25] Li, L.H. et al. (2019) VisualBERT: A simple and performant baseline for vision and language, arXiv.org. Available at: https://arxiv.org/abs/1908.03557.

[26] Jabbar, M.S., Shin, J. and Cho, J.-D. (2022) Ai Ekphrasis: Multi-modal learning with foundation models for fine-grained poetry retrieval, MDPI. Available at: https://www.mdpi.com/2079-9292/11/8/1275.

[27] Radford, A. et al. (2021) Learning transferable visual models from Natural Language Supervision, arXiv.org. Available at: https://arxiv.org/abs/2103.00020.

[28] Raffel, C. et al. (2020) Exploring the limits of transfer learning with a unified text-to-text transformer, arXiv.org. Available at: https://arxiv.org/abs/1910.10683.

[29] Tiedemann, J. and Thottingal, S. (2020) Opus-mt – building open translation services for the world, ACL Anthology. In Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, pages 479–480, Lisboa, Portugal. European Association for Machine Translation. Available at: https://aclanthology.org/2020.eamt-1.61.

[30] Sutskever, I., Vinyals, O. and Le, Q.V. (2014) Sequence to sequence learning with Neural Networks, arXiv.org. Available at: https://arxiv.org/abs/1409.3215.

[31] Dai, Z., Xie, Q. and Hovy, E. (2018) From credit assignment to entropy regularization: Two new algorithms for neural sequence prediction, arXiv.org. Available at: https://arxiv.org/abs/1804.10974.

[32] Elliott, D. et al. (2016) Multi30k: Multilingual English-German image descriptions, arXiv.org. Available at: https://arxiv.org/abs/1605.00459.

[33] Papineni, K. et al. (2001) 'Bleu', Proceedings of the 40th Annual Meeting on Association for Computational Linguistics&nbsp; - ACL '02 [Preprint]. doi:10.3115/1073083.1073135.

[34] Denkowski, M. and Lavie, A. (2014) 'Meteor Universal: Language specific translation evaluation for any target language', Proceedings of the Ninth Workshop on Statistical Machine Translation [Preprint]. doi:10.3115/v1/w14-3348.