Direct Preference Optimization on FLAN T5 for English to Romanian translation

Popescu Luca Alexandru[a], Vasilescu Andreea[b]

[a] *University of Bucharest, Academiei Street, no. 14, Bucharest, 077131, Romania*
[b] *University of Bucharest, Academiei Street, no. 14, Bucharest, 077131, Romania*

Abstract

In this project we try leveraging the Transformer Reinforcement Learning library of fine-tune an open-source model from Hugging Face, aiming to replicate and evaluate the Direct Preference Optimization on the Text-to-Text Transfer Transformer(FLAN T5) model, specifically focusing on its application in English-to-Romanian translation without diacritics.

Keywords: Machine Translation, T5, DPO

1. Introduction

Machine Translation represents the process of automated process of translating text or speech from a language to another without using computational algorithms. While advancements in machine translation have improved its accuracy, researchers are still facing challenges in capturing nuanced meanings or cultural nuances.

2. State of the ART

The following section will provide an overview of the current advancements and achievements in the Machine Translation field, as well as state of the art models, methodologies, limitations and potential applications. The purpose of this section aims to set a foundation for the conducted study that will be presented.

2.1. Different approaches of Machine Translation

Machine Translation evolved through different approaches, such as rule-based methodologies, statistical methods that analyze large bilingual corpora, or neural approaches such as transformers.

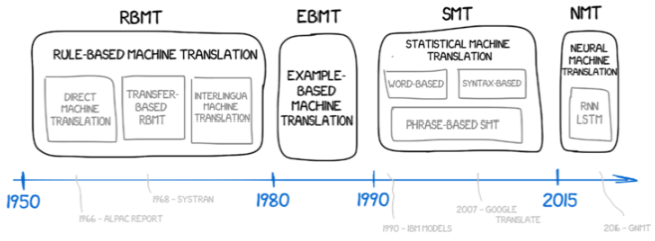


Figure 1: A brief history of Machine Translation**?**

Rule-based methodologies are hard coded linguistic rules based on grammatical structure and pre-defined linguistic rules to generate translations.

There are three types of Rule Based Machine Translation:

- Direct Systems: mapping input to output using basic rules
- Transfer RBMT Systems that employ morphological and syntactical analysis
- Interlingual RBMT Systems that use an abstract meaning.

The main approach of Rule Based Machine Translation is to link the structure of a given input sentence with the structure of the output sentence.

Minimum requirements are as follows:

- A dictionary that will map each word from language 1 to language 2.
- Rules for regular sentence structure in language 1.
- Rules for regular sentence structure in language 2.

language 1 - language we want to translate from ; language 2 - target translation language.

Example Based Machine Translation uses bilingual corpus with parallel texts.

Statistical Machine Translation involve analyzing large bilingual corpora to probabilistically model the relationship between words and phrases in different languages. The system will make translation based on learned patterns.

Neural Networks in machine translation are used to predict the likelihood of a sequence of words and they require a lot of high quality data.

2.2. *Transformers*

are a type of neural network architecture introduced in NLP that was introduced in 2017 in a paper entitled "Attention is all you need". It was revolutionary for Machine Translation tasks and various language-related tasks due to its ability to capture contextual information.

Transformers capture the relationship between words in a sequence using self-attention mechanisms, and long-range dependencies in input data.

Self Attention mechanism is the key innovation of transformers as it allows the model to focus on different parts of the input sequence and being able to capture the importance of each word in the context of the entire sentence.

Encoder-Decoder Architecture are used for Sequence-to-Sequence tasks. The encoder process the input sequence and the the decoder generates the output sequence, and both of them consist of multiple layers of self-attention mechanisms.

Attention Heads & Layers. As previously mentioned, self-attention consist of several layers of self-attention to allow the model to focus on different aspects of the input simultaneously. The model will capture hierarchical and complex patterns.

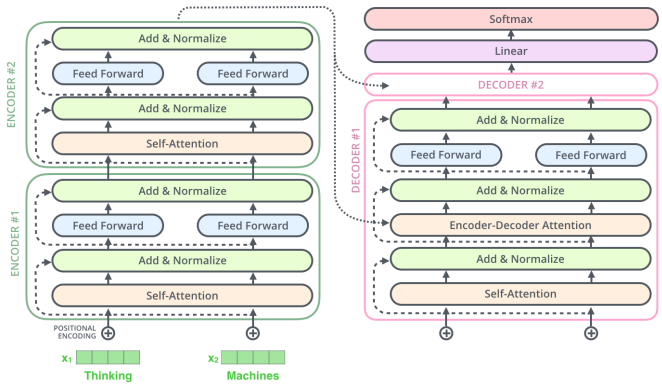
Positional Encoding are added to the input embeddings to provide information about the position of each token in the sequence.

Feed-Forward Neural Networks each attention head is followed by a feed forward neural network that will process the information extracted by the attention mechanism.

Layer Normalization and Residual Connections are used for stabilizing the training. These components will help with to prevent the vanishing gradient problem and improve the flow of information through the network.

Training: Transformers are using backpropagation and optimization algorithms that are often pretrained on large datasets and pre-trained for specific tasks.

Transfer learning: Transformers are known for their effectiveness in transfer learning. Pretrained models on large datasets can be fine-tuned for downstream tasks with smaller datasets, resulting in improved performance.



2.3. *Direct Preference Optimization (DPO)*

Direct Preference Optimization was introduced as a sophisticated approach to achieve precise control over Large Language Models (LLMs). This development was closely tied to advancements in Reinforcement Learning from Human Feedback (RLHF). RLHF represents a paradigm shift in training NLP models. In the context of LLMs, RLHF involves an interactive training process. The core idea is to train a Reward Model, which essentially acts as a surrogate for human judgment. This Reward Model is constructed based on an array of human feedback, encompassing various aspects of language generation. The training process involves presenting human evaluators with specific outputs from the language model, and then gathering their feedback. This feedback is then used to train the Reward Model, enabling it to estimate the human desirability of a given output. Once the Reward Model is sufficiently trained, it's integrated into the training of the language model using Proximal Policy Optimization (PPO). PPO, a state-of-the-art algorithm in reinforcement learning, allows for efficient and effective policy updates. Then, the language model's behavior is gradually aligned with the preferences encapsulated in the Reward Model. However, one of the primary complexities lies in the consistency of the human feedback. The accuracy of the Reward Model hinges on how well the human feedback represents a wide range of linguistic nuances and preferences. Furthermore, the use of PPO, while effective, introduces its own set of challenges. The alignment process can be unstable, particularly when dealing with ambiguous or conflicting human feedback. This instability can lead to unpredictable or undesirable behavior in the language model, posing significant challenges in maintaining consistent performance. Recognizing these complexities and instability issues inherent in RLHF, Direct Preference Optimization (DPO) was proposed as a more refined method.

DPO, on the other hand, treats the constrained reward maximization problem as a classification problem on human preference data. This approach is stable, efficient, and computationally lightweight. It eliminates the need for reward model fitting, extensive sampling, and hyperparameter tuning. **?**

Direct Preference Optimization (DPO), introduces a new parameterization of the reward model in reinforcement learning from human feedback (RLHF), allowing us to extract the optimal policy in closed form, resulting in a stable, performant, and computationally lightweight algorithm that surpasses existing methods in fine-tuning LMs to align with human preferences, particularly excelling in sentiment control and maintaining or improving response quality in summarization and single-turn dialogue, all while being simpler to implement and train than PPO-based RLHF. **?**

DPO directly defines the preference loss as a function of the policy, instead of training a reward model first. During the fine-tunining phase, the LLM is used as a reward model, optimizing the policy using a binary cross-entropy objective, leveraging on human preference data to determine which responses are prefered and which are not. **?**

2.4. *How does DPO work?*

The DPO pipeline can be broken down into two main stages:

- **Supervised Fine-Tuning (SFT)**: This is the initial step where the model is fine-tuned on a dataset or datasets of interest. It involves adjusting the model's parameters to better align with specific task requirements and data characteristics.
- **Preference Learning**: After SFT, the model undergoes preference learning. This stage utilizes preference data, which ideally comes from the same distribution as the SFT examples. This data helps the model learn to prioritize outputs that align more closely with human preferences.

Supervised Fine-Tuning (SFT) serves as the foundational step in the DPO process. It is a targeted method where a Large Language Model (LLM) is further trained on a labeled dataset. This fine-tuning process provides the model with a clear mapping between specific inputs and the desired outputs, ensuring that it can accurately generate responses that are relevant to the specific tasks at hand.

SFT refines the model's output to ensure that the outputs are not only accurate but also appropriate and consistent with the task's context. This step is crucial for setting a strong baseline for the model's performance, upon which preference learning can build.

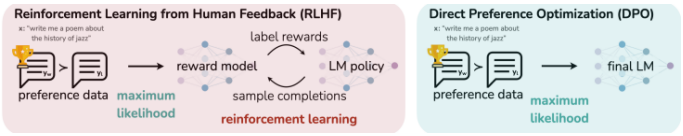


Figure 2: DPO optimizes for human preferences while avoiding RL **?**

2.5. Text-to-Text Transfer Transformer

T5, or Text-To-Text Transfer Transformer, is a transformer-based neural network architecture for natural language processing (NLP) developed by researchers at Google to serve as an open-source foundational model. It was introduced in the paper entitled "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer".

T5 is designed to handle a wide range of NLP tasks by framing them all as a text-to-text problem. Instead of having task-specific architectures, T5 is trained on a diverse set of tasks by converting each task into a text generation problem. This means both input and output are treated as text sequences, and the model is trained to generate the target text given the input text.

The flexibility of T5 allows it to be fine-tuned on various downstream tasks such as summarization, question answering, translation, and more. The model has demonstrated strong performance across different benchmarks and has contributed to the success of transfer learning in NLP.

Fig 1: A diagram of text-to-text framework. Every task we consider—including translation, question answering, and classification—is cast as feeding our model text as input and training it to generate some target text. This allows to use the same

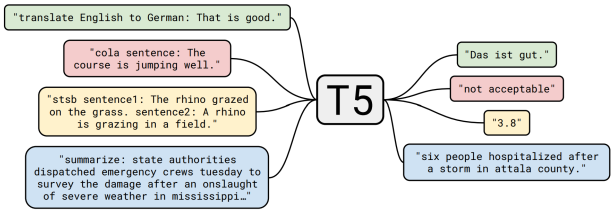


Figure 3: Figure 1

model, loss function, hyperparameters, etc. acros diverse set of tasks. It also provides a standard testbed for the methods included in empirical survey **?**

3. Experiment

In the following are going to explain our work done on our project entitled DPO on T5, as well as present how the experiment was conducted. This chapter contains information regarding implementation of the coding part that stands behind this paper, what we have tried and how we got to the final version.

3.1. Papers our project is based on

For this project we have two main papers that were a source of inspiration for us:

- FLAN T5-model Hugging Face page **?**
- Direct Preference Optimization repository **?**

3.2. Tools used

For this experiment,we used **Python** programming language.

3.3. Results

The project requires leveraging DPO in a machine translation context, as opposed to the chat preference one DPO is usually employed. To accommodate this, we will be trying to steer the model into translating English text to Romanian without using diacritics (as opposed to its original form, where it is using diacritics). We used the transformer TRL library to reproduce the DPO as a simple RL-free algorithm for training the T5 model from preferences.

This task was chosen due to the lack of datasets containing the (prompt, chosen, rejected) format in a machine tranlsation context, as transliteration can be automated easily and efficiently. Since FLAN T5 is a small model with a small context (the ones used in the experiments are: T5-small: 74M params and T5-base: 248M params) we chose a dataset (the Tapaco dataset) containing small and simple English sentences ("prompt") to be translated by the model (the "rejected" ones) and then transliterated ("chosen").

Afterwards, DPO is applied to the model. A full parameter retraining and a QLoRA variant were developed, neither of which seem to yield good results.

We first attributed this to the way the T5 pipeline tokenizes data, likely grouping the words into larger chunks and, thus making it harder for the model to discover a transliteration mapping. A similar experiment was performed on a punctuation removal from translation task, as punctuation is guaranteed to be a separate token, which should just be removes, as opposed to mapping. Unfortunately, this experiment also yielded poor results. There are a number of instances in forums reporting the same strange behaviour when trying to to DPO on smaller scale models.

The code provided supports several models- so we added the T5 model from Hugging face.

4. Conclusions

We have provided a basic pipeline for applying DPO on language translation tasks, but unfortunately were unable to train a large enough model to achieve a satisfying result. We've also introduced two possible tasks a language translation model could be optimized on in the context of English to Romanian translation, since open-source data is unavailable.

Issues encountered

1. Surprising lack of resources and information for applying DPO on smaller (¡7B) models.
2. A pipeline for fine-tuning a Mistral 7B model was created for our use-case, but we were not able to fit into less than 16Gb of VRAM.
3. Even with quantization, video memory requirements represent a problem since two copies of the model need to be loaded.
4. We found a few discussions online about other people encountering the same problem as us for the T5 model, but details were scarce.

5. Future Work

Running the pipeline for a bigger model could potentially yield interesting results, as 7B parameters is the low end of where we've seen good published results on DPO, but unfortunately we did not have access to higher end hardware at the time.