Direct Preference Optimization on Text-to-Text Transfer Transformer(T5)

Popescu Luca Alexandru & Vasilescu Andreea

University of Bucharest
Faculty of Mathematics and Informatics
Masters of Natural Language Processing

January 2024

Table of Contents

Table of Contents

Introduction

Aim of the project

In this project we leveraged on tone GitHub repository and model from Hugging Face, aiming to replicate and evaluate the Direct Preference Optimization on Text-to-Text Transfer Transformer, specifically focusing on its application in English-to-Romanian translation without diacritics.
For this project we have two main papers that were a source of inspiration for us:

- T5-model from hugging-face
- Direct Preference Optimization git

Introduction to Machine Translation

About

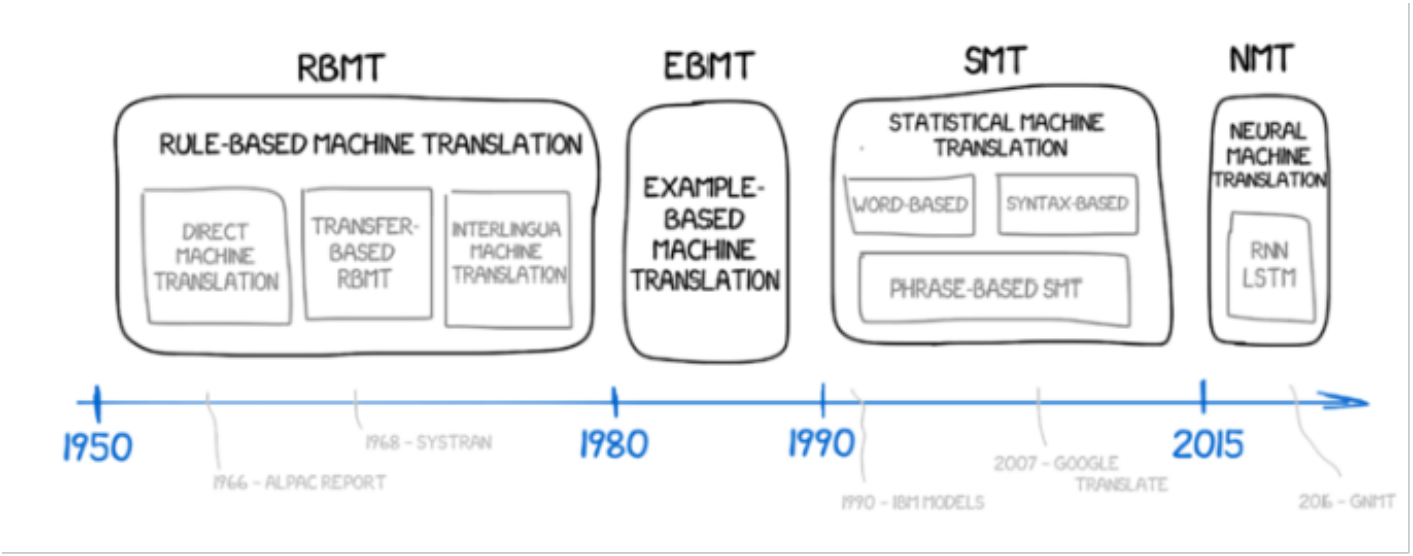


Figure: A brief history of Machine Translation

Introduction to Machine Translation

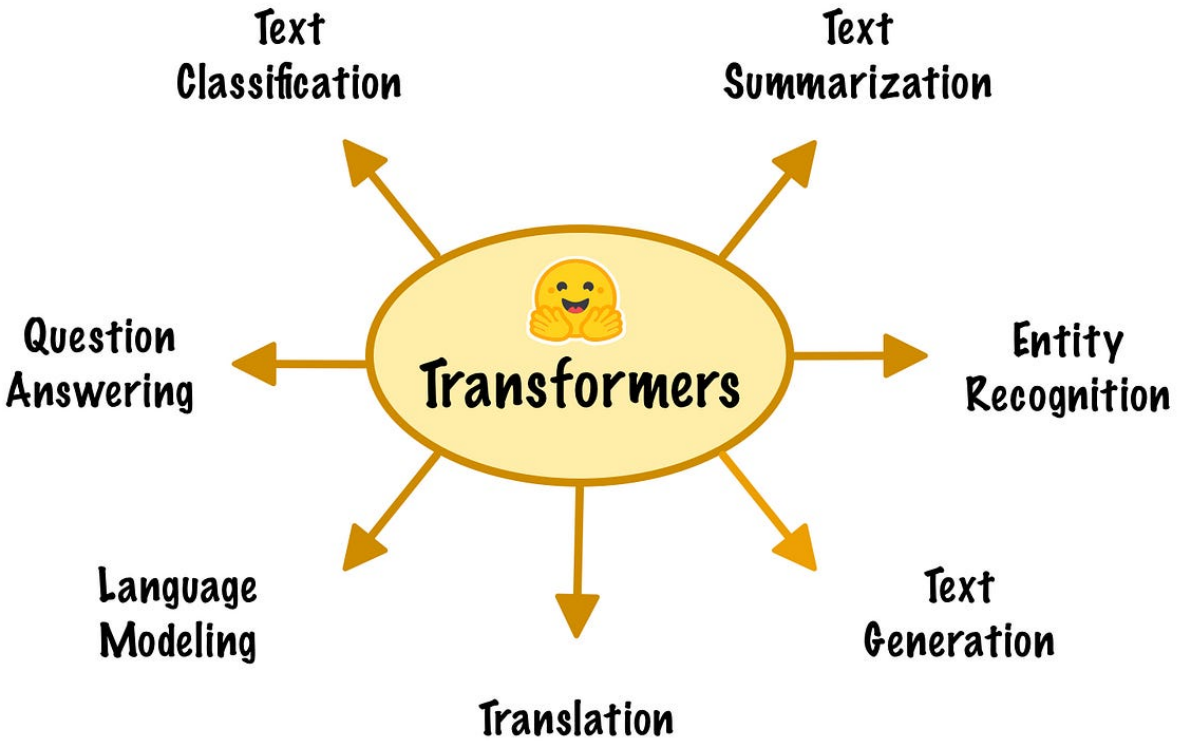
Different approaches of Machine Translation

- **Rule Based Machine Translation** are hard coded linguistic rules,
- **Example Based Machine Translation** uses bilingual corpus with parallel texts,
- **Statistical Machine Translation** involve analyzing bilingual corpora to probabilistically model the relationship between two words or phrases in different languages,
- **Neural Machine Translation** used to predict the likelihood of a sequence of words.

Transformers

What are transformers?

Transformers are a type of NN architecture that is used in Machine Translation and various language related tasks as it captures the relationship between words in a sequence using self-attention mechanism, and long range dependencies in input data.

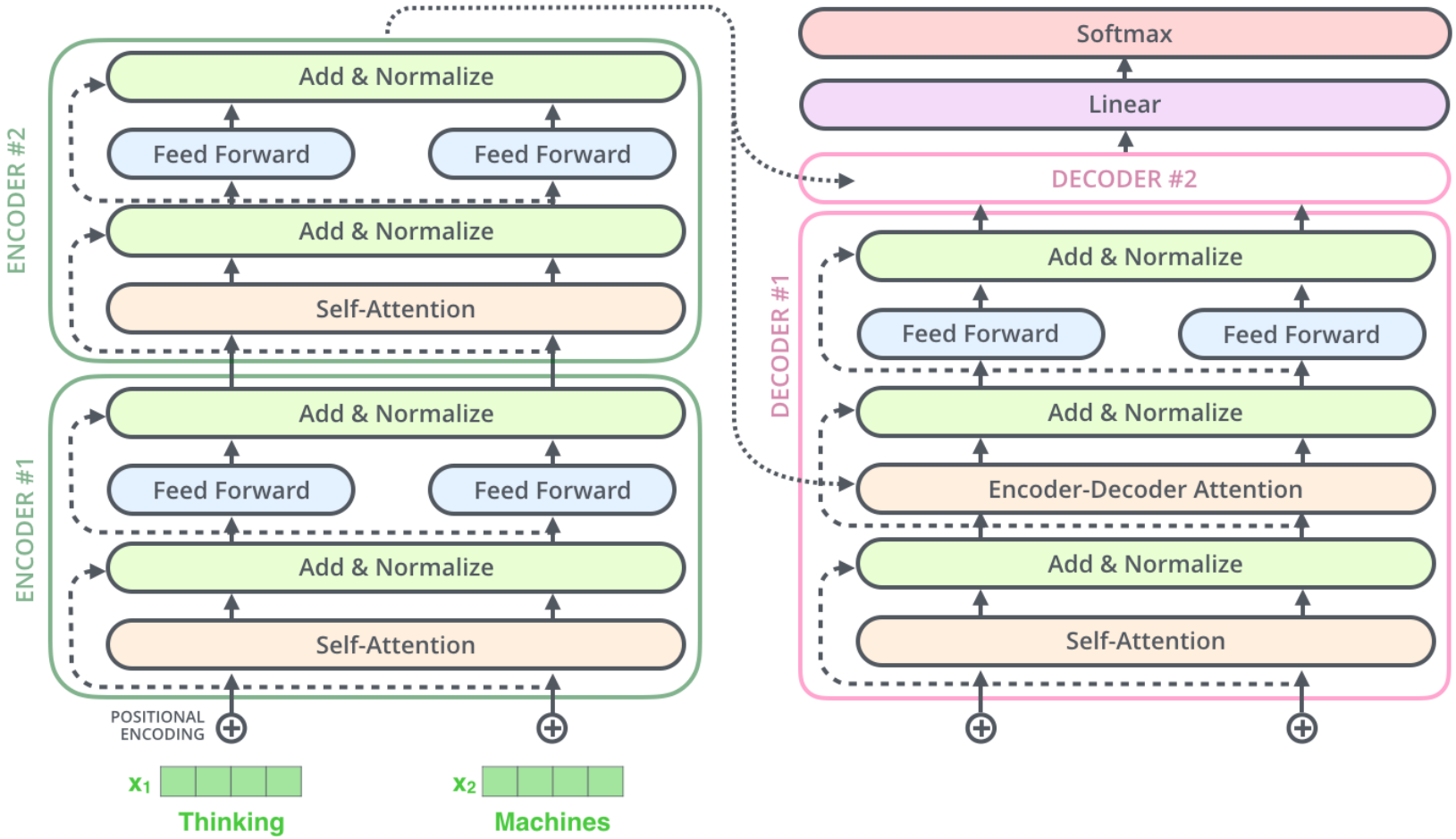


Transformers

Self-attention

Transformers are a type of NN architecture that is used in Machine Translation and various language related tasks as it captures the relationship between words in a sequence using self-attention mechanism, and long range dependencies in input data.
Self Attention is the key mechanism of transformers as it allows the model to focus on different parts of the input by assigning different weights to different words in the input sentence, so it captures the importance of each word in the context.

Transformers



Transformers

How do transformers work?

- **Self-Attention Mechanism**: giving weights to different words in a sentence,
- **Encoder-Decoder Architecture**: for sequence-to-sequence tasks, encoder generates the input sequence and decoder generates the output sequence,
- **Attention Heads & Layers**: enables the model to capture hierarchical and complex patterns,
- **Positional Encoding** added to input embeddings to provide information on token position,
- **Feed-Forward Neural Networks** will process the information extracted by attention mechanism,
- **Layer Normalization and Residual Connections** are used for stabilizing the training,
- **Training**: using backpropagation and usually pre-trained algorithms.
- **Transfer learning**: pre-trained model can be fine-tuned for downstream taskes with smaller datasets.

What is DPO?

- DPO is a method introduced to achieve precise control over LLMs.
- Reinforcement Learning from Human Feedback was based on a Reward Model using Proximal Policy Optimization, but it was unstable.
- DPO treats the contstrained reward maximization as a classification problem on human preference data, as a stable approach, eliminating the reward model fitting.

How does DPO work?

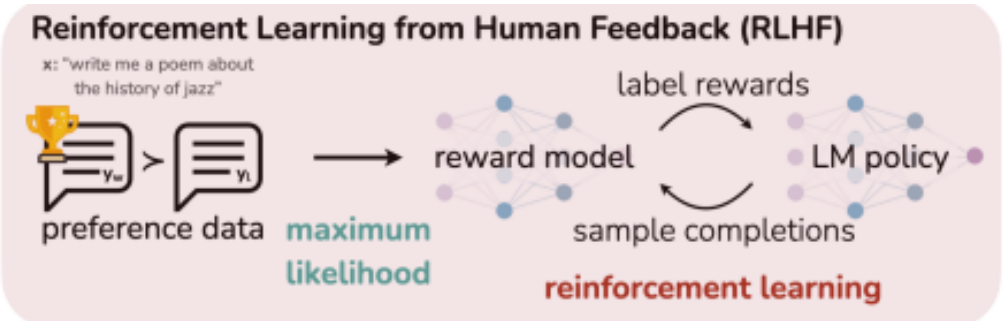
There are two main stages of DPO:

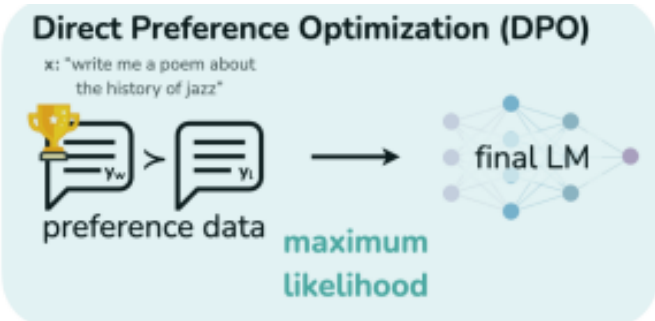
- **Supervised Fine-tuning (SFT)**: initial step where the model is fine-tuned on the dataset/datasets of interest.
- **Preference Learning**: After SFT, the model undergoes preference learning using preference data, ideally from the same distribution as the SFT examples.

During fine-tuning phase the LLM is being used as a reward model, leveraging on human preference data to determine weather a response is prefered or not.
This way, there is no prerequisite of training a reward model.

DPO

DPO 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.





Supervised Fine-Tuning

- first step of DPO
- LLM is further trained on a labeled dataset
- there is a clear mapping between specific inputs and desired outputs
- SFT refines the output of the model to be accurate, appropriate & consistent.

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".

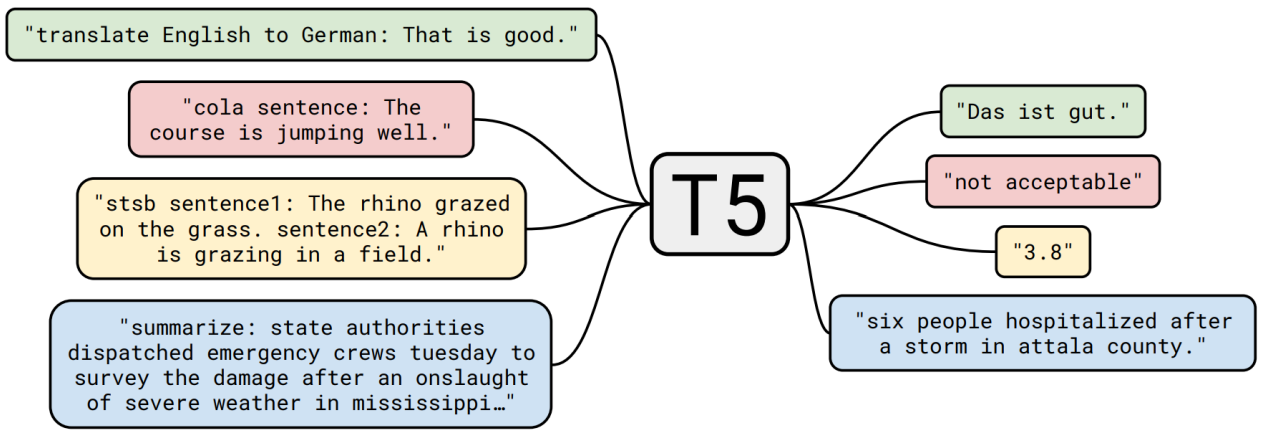


Figure: Figure 1

We have reproduced the paper entitled to directly optimize a language model to adhere to human preferences, without explicit reward modeling or reinforcement learning using DPO git repro.
The aim was to use a simple RL-free algorithm for training the T5 model from preferences.
The code provided supports several models- so we added the T5 model from Hugging face.

DPO pipeline had two stages:

- Run suprtvised fine-tuning(SFT) on the dataset.
- Run preference learning on the model from step 1, using preference data from the same distribution as the SFT.

Conclusions

We have provided a basic pipeline for applying DPO on lan- guage 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.