Finetuning smaller open source LLM's for conversation in Shakespearean English

Documentation

Roudranil Das
roudranil@cmi.ac.in

Contents

1 Introduction

Presently, our primary objective is to meticulously refine a chat model, enabling it to engage in conversations with us using Shakespearean English, notwithstanding our prompts being in plain English. The initial phase involves experimentation with smaller models, specifically in the range of 1 billion to 7 billion parameters.

2 Setup

Execute the following commands to initialize the environment.

```
python -m venv .env
source .env/scripts/bin/activate
pip install -r requirements.txt
```

3 Dataset Creation

3.1 Overview

A swift internet search reveals two primary sources for datasets containing works of Shakespeare:

- Tiny Shakespeare from Andrej Karpathy's repository [🤗 Datasets]
- Translation of complete works of Shakespeare - Shakescleare [Kaggle datasets]

However, for my specific purpose, I required a dataset containing both a dialog and its translation, along with the response pair of the original dialog and translation. The tiny Shakespeare dataset lacks translations, and the translated dataset does not have dialogs in the correct sequence. Instead of devising a workaround, I opted to perform web scraping.

3.2 Scraping

The code for scraping can be found in scrapy.py, inspired by shakespeare_crawler.py from ToruOwO/style-transfer-writing. The scraping is conducted in a very specific format, with the data directory structured as follows:

```
data
|---book_name
|   |---chapter_name.json
.   .
.   .
.   .
```

Each JSON file has the following structure:

```
{
    "dialogs":[
        {
            "original": "original dialog here",
            "translated": "translated dialog here"
        },
        {
            // more dialog pairs
        },
    ]
}
```

Note that the array of dialogs maintains the same order as they appear in the books, signifying that any dialog can be considered the response to the dialog immediately preceding it in the array. To run the scraper, navigate to the src directory and run

```
python scraping.py
```

3.3 Processing to csv

This part is done in dataset.py. Simply the JSON files are laoded then they are eported to a csv with the following columns

id	translated_dialog	og_response
bookname-chaptername-line-#	some dialog in modern english	reply to that dialog as written in shakespearean english

To create the datasets navigate to the `src` directory and run

```
python dataset.py
```

The final dataset is available as Shakespearean and Modern English Conversational Dataset on 🤗 Datasets.

4 Models and finetuning

4.1 Selected models

I aimed to finetune the following models

Model name	Size
ericzzz/falcon-rw-1b-instruct-openorca	$1B$
togethercomputer/RedPajama-INCITE-Chat-3B-v1	$3B$
mistralai/Mistral-7B-Instruct-v0.2	$7B$

4.2 Model Performance

4.2.1 ericzzz/falcon-rw-1b-instruct-openorca

Training Information:

Currently, the model has been trained for a total of 6600 steps, reaching a minimum loss of 1.8.

Prompt Template:

`f"<SYS> {system} <INST> {user} <RESP> {response}"`

4.2.2 togethercomputer/RedPajama-INCITE-Chat-3B-v1

Training Information:

Important: The model requires extended training duration.

The model has been trained for 500 steps, achieving a minimum loss of 2.29.

Prompt Template:

`f"<system>: {system}\n<human>: {user}\n<bot>: {response}"`

4.2.3 mistralai/Mistral-7B-Instruct-v0.2

Training Information:

Important: The model requires extended training duration.

The model has been trained for 2500 steps. A decision was made to extend the training on a larger model to explore its capabilities. However, further extension may lead to Colab runtime disconnection.

Prompt Template:

`f"<s>[INST] {system}\n\n{user} [/INST] {response} </s>"`