Fine-Tuning GPT-2 for Bengali Auto Text Completion Using Literary Corpus: A Qualitative Analysis

Taseen Mubassira
Mehedi Hasan
Supervised By: Dr. Mohammed Eunus Ali

The Model

- OpenAI GPT-2 model was proposed in Language Models are Unsupervised Multitask Learners paper

- Original GPT2 model was a causal (unidirectional) transformer pretrained using language modeling on a very large corpus of ~40 GB of text data

- This model has the same configuration but has been pretrained on bengali corpus of mC4 (multilingual C4) dataset

- mC4 dataset: https://huggingface.co/datasets/mc4/viewer/bn/train

Running the Model

```
1 from transformers import pipeline
2
3 generator = pipeline('text-generation',model="flax-community/gpt2-bengali", tokenizer='flax-community/gpt2-bengali')
```

```
Downloading (…)lve/main/config.json: 100% 864/864 [00:00<00:00, 6.82kB/s]
Downloading (…)"pytorch_model.bin";: 100% 510M/510M [00:08<00:00, 54.8MB/s]
Downloading (…)/main/tokenizer.json: 100% 1.76M/1.76M [00:00<00:00, 8.04MB/s]
```

Running the Model: Results

```
1 print(generator("সড়ক দুর্ঘটনায় ৪ জন ব্যক্তি নিহত"))
```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
[{'generated_text': 'সড়ক দুর্ঘটনায় ৪ জন ব্যক্তি নিহত হয়েছেন। নিহতরা হলেন- বানিয়াচং উপজেলার ম'}]
```

```
1 print(generator("সড়ক দুর্ঘটনায় ৪ জন"))
```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
[{'generated_text': 'সড়ক দুর্ঘটনায় ৪ জন নিহত ।।সিনহাকে গ্রেপ্তারের পর প্রতিবাদ করা না হলে কঠো'}]
```

Running The Model: Results

```
1 print(generator("রিক্সা ভাড়া কমানোর জন্য রিকশাওয়ালার সঙ্গে"))
```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
/usr/local/lib/python3.8/dist-packages/transformers/generation/utils.py:1273: UserWarning: M
  warnings.warn(
[{'generated_text': 'রিক্সা ভাড়া কমানোর জন্য রিকশাওয়ালার সঙ্গে আলোচনায় বসলেন। অনেকেই টুরি'}]
```

```
1 print(generator("বাজারে জংলি মোরগ"))
```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
/usr/local/lib/python3.8/dist-packages/transformers/generation/utils.py:1273: UserWarning: Neithe
  warnings.warn(
[{'generated_text': 'বাজারে জংলি মোরগ ও চিটাগাংয়ের এক যুবককে অজ্ঞান অবস্থায় পাওয়া গেছে বলে জ'}]
```

Limitation of Current Model

- The training dataset is largely composed of newspaper articles
- As a result, the predictions are not always relevant

```
1 print(generator("রহমান সাহেব বাজারে এসেছেন মাছ "))
```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
[{'generated_text': 'রহমান সাহেব বাজারে এসেছেন মাছ বুয়েটেও সিরাজুল স্যারের ছাত্ররা'}]
```

- Gives news article oriented predictions

```
1 print(generator("ডিমের দাম বেড়ে গেছে তাই ডিম কিনতে"))
```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
[{'generated_text': 'ডিমের দাম বেড়ে গেছে তাই ডিম কিনতেও ভোগান্তি পোহাতে হচ্ছে ক্রেত'}]
```

Fine Tuning and Other Modifications

- Train the model on different corpus
- Examples: bengali literature (novels, songs)
- Experiment on different parameters

Steps to Fine-Tune the model

1. **Prepare the dataset**
   a. This involves converting the dataset into text files and adding it into the corpus
2. **Tokenize the dataset**
   a. The dataset has to be tokenized with the same tokenizer that was used to training the original model
3. **Fine-tune the GPT-2 Model**
   a. The `Trainer` class from the `Transformers` library provides an easy to use interface to train the model, including options for configuring the training hyperparameters
4. **Generate text with fine-tuned model**
   a. Once the GPT-2 model have been trained (fine-tuned) on the custom dataset, the `pipeline` interface can be used to get the predictions of bengali text

Train the Model on different corpus

- We picked some literature content and introduced it into the existing training corpus

- Literatures included:
  - Bengali Novels
  - Bengali Songs

Results after Fine-Tuning

```python
# Define the text generation pipeline
generator = pipeline('text-generation', model=model, tokenizer=tokenizer)

# Generate text given an input prompt
prompt = 'আমি আমার আমিকে'
output = generator(prompt, max_length=100, do_sample=True, temperature=0.7)

# Print the generated text
print(output[0]['generated_text'])
```

```
Generate config GenerationConfig {
        'bos_token_id': 50256,
        'do_sample': true,
        'eos_token_id': 50256,
        'max_length': 50,
        'transformers_version': "4.26.1"
}

Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
আমি আমার আমিকে চিরদিন এই বাংলায় খুজে পেয়েছি তাই অনেকেই বলেছেন আমার নাম মুজিবুর রাহমান
```

Results after Fine-Tuning: Changing max_length

```
1 prompt = 'আমি আমার আমিকে'
2 output = generator(prompt, max_length=50, do_sample=True, temperature=0.7)
3 print(output[0]['generated_text'])
```

```
Generate config GenerationConfig {
  "bos_token_id": 50256,
  "do_sample": true,
  "eos_token_id": 50256,
  "max_length": 50,
  "transformers_version": "4.26.1"
}

Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
আমি আমার আমিকে খুঁজছি কিন্তু খুঁজে পাচ্ছি না কেন? আমি কি আমার সাথে থাকতে
```

Results after Fine-Tuning: Changing max_length

```
1 prompt = 'আমি আমার আমিকে'
2 output = generator(prompt, max_length=15, do_sample=True, temperature=0.7)
3 print(output[0]['generated_text'])
```

```
Generate config GenerationConfig {
  "bos_token_id": 50256,
  "do_sample": true,
  "eos_token_id": 50256,
  "max_length": 50,
  "transformers_version": "4.26.1"
}

Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
আমি আমার আমিকে চিনি যে
```

Findings

- Adding different literature in the corpus changes the quality of predictions
- Changing epoch improves the quality of predictions
- However, an epoch over 50 starts giving the same prediction everytime (overfitting)
- Changing max_length improves the quality of predictions
- The context starts getting derailed with longer predictions

Thank You!