Gen AI Lab 1

Name: Ayush Chakraborty
SRN: PES2UG23CS112
Date: 20/1/26

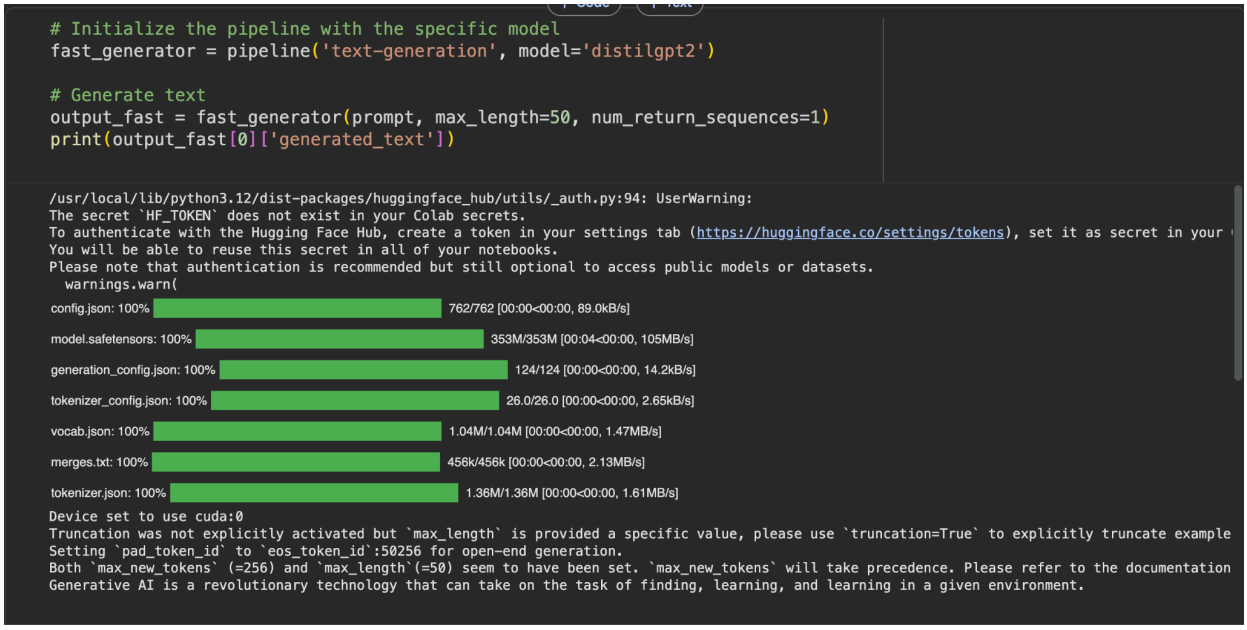
Observations:

Transformers library acts as a chain between the code and the models on hugging face. In this library, pipeline() function exists to do these tasks: preprocessing, inference, postprocessing. pipeline('text-generation', model="...") does this.

The document given is just a standard text containing info about genAI

We look at two models from huggingface. One is distilGPT2 which is not very coherent in its inference, and the other is gpt2, which is smaller but more coherent

With seed of 42:



```
# Initialize the pipeline with the specific model
fast_generator = pipeline('text-generation', model='distilgpt2')

# Generate text
output_fast = fast_generator(prompt, max_length=50, num_return_sequences=1)
print(output_fast[0]['generated_text'])
```

```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
config.json: 100% 762/762 [00:00<00:00, 89.0kB/s]
model.safetensors: 100% 353M/353M [00:04<00:00, 105MB/s]
generation_config.json: 100% 124/124 [00:00<00:00, 14.2kB/s]
tokenizer_config.json: 100% 26.0/26.0 [00:00<00:00, 2.65kB/s]
vocab.json: 100% 1.04M/1.04M [00:00<00:00, 1.47MB/s]
merges.txt: 100% 456k/456k [00:00<00:00, 2.13MB/s]
tokenizer.json: 100% 1.36M/1.36M [00:00<00:00, 1.61MB/s]
Device set to use cuda:0
Truncation was not explicitly activated but `max_length` is provided a specific value, please use `truncation=True` to explicitly truncate example
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Both `max_new_tokens` (=256) and `max_length`(=50) seem to have been set. `max_new_tokens` will take precedence. Please refer to the documentation
Generative AI is a revolutionary technology that can take on the task of finding, learning, and learning in a given environment.
```

```
smart_generator = pipeline('text-generation', model='gpt2')

output_smart = smart_generator(prompt, max_length=50, num_return_sequences=1)
print(output_smart[0]['generated_text'])
```

```
config.json: 100% 665/665 [00:00<00:00, 68.0kB/s]
model.safetensors: 100% 548M/548M [00:02<00:00, 283MB/s]
generation_config.json: 100% 124/124 [00:00<00:00, 10.4kB/s]
tokenizer_config.json: 100% 26.0/26.0 [00:00<00:00, 1.39kB/s]
vocab.json: 100% 1.04M/1.04M [00:00<00:00, 1.22MB/s]
merges.txt: 100% 456k/456k [00:00<00:00, 784kB/s]
tokenizer.json: 100% 1.36M/1.36M [00:00<00:00, 2.01MB/s]
Device set to use cuda:0
Truncation was not explicitly activated but `max_length` is provided a specific value, please use `truncation=True` to explicitly truncate examples
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Both `max_new_tokens` (=256) and `max_length`(=50) seem to have been set. `max_new_tokens` will take precedence. Please refer to the documentation
Generative AI is a revolutionary technology that enables a wide range of intelligent systems to work independently from one another. It introduces

In this article, we will discuss the main features of the new AI platform, and how it can be used to help us create a world that will improve our li

1. How Can I Use It?

The concept of AI is not new. It has been used by many people to measure their mental health and health-related behaviors, and as a tool for medical

It is based on the premise that AI is a way for humans to move towards a more efficient way of thinking, and therefore, a better way of living.

In this article, we will explain what AI can do.

What does it do

In this article, we will explain how all of our cognitive and emotional systems interact with the AI platform. The main features of AI are:

A new way of thinking about AI
```

```
questions = [
    "What is the fundamental innovation of the Transformer?",
    "What are the risks of using Generative AI?"
]

for q in questions:
    res = qa_pipeline(question=q, context=text[:5000])
    print(f"\nQ: {q}")
    print(f"A: {res['answer']}")
```

```
Q: What is the fundamental innovation of the Transformer?
A: to identify hidden patterns, structures, and relationships within the data

Q: What are the risks of using Generative AI?
A: data privacy, intellectual property, and academic integrity
```

Now for the seed of 67, the effects would be that the variance of the inference would change a bit, but the context remains the same.

Step 3: Fast Model (`distilgpt2`)

Let's see how the smaller model performs.

```
# Initialize the pipeline with the specific model
fast_generator = pipeline('text-generation', model='distilgpt2')

# Generate text
output_fast = fast_generator(prompt, max_length=50, num_return_sequences=1)
print(output_fast[0]['generated_text'])
```

```
Device set to use cuda:0
Truncation was not explicitly activated but `max_length` is provided a specific value, please use `truncation=True` to explicitly truncate examples
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Both `max_new_tokens` (=256) and `max_length`(=50) seem to have been set. `max_new_tokens` will take precedence. Please refer to the documentation f
Generative AI is a revolutionary technology that enables the creation of autonomous robots on the road. In the future, the technology will be able t
```

The later part of the inference differs, with seed 42, “....that can take on the task of …” and with seed of 67, it is “...that enables the creation of autonomous…”. So this is the impact of seed

Step 4: Standard Model (`gpt2`)

Now let's try the standard model.

```
smart_generator = pipeline('text-generation', model='gpt2')

output_smart = smart_generator(prompt, max_length=50, num_return_sequences=1)
print(output_smart[0]['generated_text'])
```

```
C:\Users\piyus\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\huggingfa
To support symlinks on Windows, you either need to activate Developer Mode or to run Python as an administrator. In order to activate developer mode
  warnings.warn(message)
Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download. For better performance, inst
Device set to use cpu
Truncation was not explicitly activated but `max_length` is provided a specific value, please use `truncation=True` to explicitly truncate examples
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Both `max_new_tokens` (=256) and `max_length`(=50) seem to have been set. `max_new_tokens` will take precedence. Please refer to the documentation f
Generative AI is a revolutionary technology that allows users to build AI that can help solve complex problems. It brings together hundreds of diffe

The AI is a model of human intelligence, and has many aspects that are similar to artificial intelligence. It can learn from humans, and it can adap

It is the main driving force behind the new Artificial Intelligence, and the AI is very important to the success of AI. The new AI is designed to wo

The AI is designed to be scalable and adaptable to different environments. It can be used to solve complex problems without relying on humans. It ca

The new AI is designed to work out problems that need to be solved in a way that is
```

Same with the gpt2 output

Quality Summarizer

```
smart_sum = pipeline("summarization", model="facebook/bart-large-cnn")
res_smart = smart_sum(transformer_section, max_length=60, min_length=30, do_sample=False)
print(res_smart[0]['summary_text'])
```

```
Device set to use cuda:0
The introduction of the Transformer architecture in the 2017 paper "Attention is all you need" was a watershed moment in AI. It provided a more effe
```

```
questions = [
    "What is the fundamental innovation of the Transformer?",
    "What are the risks of using Generative AI?"
]

for q in questions:
    res = qa_pipeline(question=q, context=text[:5000])
    print(f"\nQ: {q}")
    print(f"A: {res['answer']}")
```

```
Q: What is the fundamental innovation of the Transformer?
A: to identify hidden patterns, structures, and relationships within the data

Q: What are the risks of using Generative AI?
A: data privacy, intellectual property, and academic integrity
```

```
snippet = text[:1000]
entities = ner_pipeline(snippet)

print(f"{'Entity':<20} | {'Type':<10} | {'Score':<5}")
print("-"*45)
for entity in entities:
    if entity['score'] > 0.90:
        print(f"{entity['word']:<20} | {entity['entity_group']:<10} | {entity['score']:.2f}")
```

```
Entity               | Type       | Score
---------------------------------------------
AI                   | MISC       | 0.98
PES University       | ORG        | 0.99
AI                   | MISC       | 0.98
Large Language Models | MISC       | 0.91
LLMs                 | MISC       | 0.90
Transformer          | MISC       | 0.99
```

```
masked_sentence = "The goal of Generative AI is to create new [MASK]."
preds = mask_filler(masked_sentence)

for p in preds:
    print(f"{p['token_str']}: {p['score']:.2f}")
```

```
applications: 0.06
ideas: 0.05
problems: 0.05
systems: 0.04
information: 0.03
```

The rest of the outputs are given too.

For the parts of speech, we get:

NN: Noun, singular or mass
VBZ: Verb, 3rd person singular present

JJ: Adjective
DT: Determiner
IN: Preposition or subordinating conjunction

All these are POS describers of the nltk library

Q) To summarise text, what is used?
A) BERT is used to summarise text, and it incorporated the encoder and decoder

Observations for different models:

Task	Model	Classification (Success/Failure)	Observation (What actually happened?)	Why did this happen? (Architectural Reason)
Generation	BERT	Failure	Generated repetitive loops, garbage text, or special tokens (e.g., `[SEP]`, `[PAD]`).	**BERT is an Encoder-only model.** It is designed to understand the full context of a sentence at once (bidirectional), not to predict the next word in a sequence (unidirectional generation).
	RoBERTa	Failure	Generated nonsensical symbols, spaces, or repetitive phrases.	**RoBERTa is also Encoder-only.** Like BERT, it optimizes for Masked Language Modeling (understanding), not for causal text generation.
	BART	Success	Generated a coherent, grammatically correct completion to the prompt.	**BART is an Encoder-Decoder model.** It has a "Decoder" component specifically designed for autoregressive text generation (predicting the next token), similar to GPT.
Fill-Mask	BERT	Success	Correctly predicted contextually relevant words (e.g., "create", "generate").	**This is BERT's native training task.** It was trained on **Masked Language Modeling (MLM)**, where it learns to fill in missing words using surrounding context.
	RoBERTa	Success	Correctly predicted relevant words (often with higher confidence than BERT).	**RoBERTa is an optimized BERT.** It uses the same MLM objective (with dynamic masking) and is highly effective at understanding context to fill gaps.
	BART	Success	Predicted relevant words to fill the mask.	**BART is trained on Text Infilling.** This objective is very similar to masking; it learns to reconstruct corrupted text, making it capable of filling in blanks.
QA	BERT	Failure / Poor	Returned random spans of text, punctuation, or empty strings that did not answer the question.	**Lack of Fine-Tuning.** This is a "Base" model. While the architecture *can* do QA, this specific version has not been fine-tuned on a QA dataset (like SQuAD) yet.
	RoBERTa	Failure / Poor	Returned random spans or unrelated text.	**Lack of Fine-Tuning.** Like BERT, the base model understands language but doesn't know the specific structure of "Question Answering" (extracting answer spans) without further training.
	BART	Failure / Poor	Returned random spans or failed to extract the answer.	**Lack of Fine-Tuning.** Even though it generates text, the base model hasn't been taught the specific task of extracting answers from a provided context.