

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%	<div></div>	665/665 [00:00<00:00, 68.0kB/s]
model.safetensors: 100%	<div></div>	548M/548M [00:02<00:00, 283MB/s]
generation_config.json: 100%	<div></div>	124/124 [00:00<00:00, 10.4kB/s]
tokenizer_config.json: 100%	<div></div>	26.0/26.0 [00:00<00:00, 1.39kB/s]
vocab.json: 100%	<div></div>	1.04M/1.04M [00:00<00:00, 1.22MB/s]
merges.txt: 100%	<div></div>	456k/456k [00:00<00:00, 784kB/s]
tokenizer.json: 100%	<div></div>	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 for more details.
Generative AI is a revolutionary technology that enables a wide range of intelligent systems to work independently from one another. It introduces a new way of thinking about AI.

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 lives.

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 diagnosis.

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"Q: {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 for more details.
Generative AI is a revolutionary technology that enables the creation of autonomous robots on the road. In the future, the technology will be able to

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\huggingface_hub\utils\logging.py:144: FutureWarning: 'warn' is deprecated and will be removed in a future version. Use 'warnings.warn(message)' instead.
Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download. For better performance, install the 'hf_xet' package.
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 for more details.
Generative AI is a revolutionary technology that allows users to build AI that can help solve complex problems. It brings together hundreds of different disciplines and technologies to create a new form of artificial intelligence. 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 adapt to new situations. 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 work out problems that need to be solved in a way that is scalable and adaptable to different environments. It can be used to solve complex problems without relying on humans. It can be used to create new products and services that are better than anything that has ever been created before. 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 effective

```
501 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 descriptors of the nltk library

Q) To summarise text, what is used?

A) BERT is used to summarise text, and it incorporates the encoder and decoder